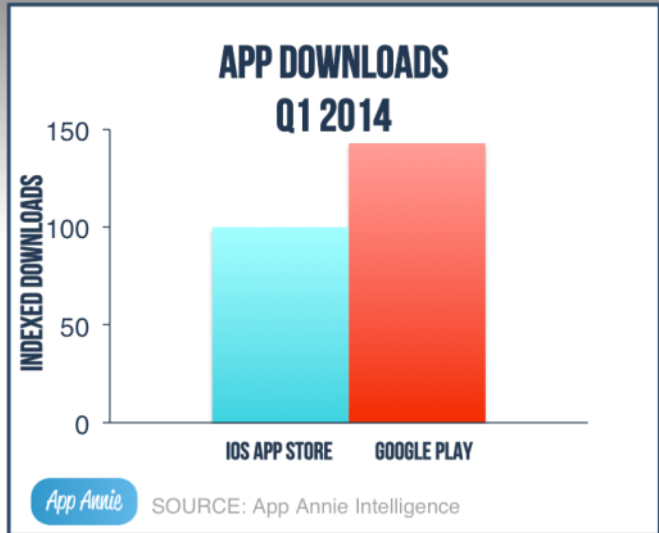# HYBRID APP DEV INTRO &
## APP DESIGN GUIDELINES

DR. JAVED KHAN

v.j.khan@tue.nl          khan.gr          @v_j_khan

CREATIVE APPS, DEC2016

TU/e

# MOBILE IS **HUGE**



APP DOWNLOADS
Q1 2014

INDEXED DOWNLOADS

150
100
50
0

IOS APP STORE    GOOGLE PLAY

App Annie    SOURCE: App Annie Intelligence

| Users of mobile apps worldwide by region 2012-2017 according to Portio Research | 2012 | 2013 | 2017 |
|---|---|---|---|
| App users worldwide | 1.2 billion | N/A | 4.4 billion |
| Asia Pacific | 30% | 32% | 47% |
| Europe | 29% | 28% | 21% |
| North America | 18% | 17% | 10% |
| Middle East & Africa | 14% | 13% | 12% |
| Latin America | 9% | 10% | 10% |
| Source: © Portio Research (March 2013) | | via: © mobiThinking | |



**HOW MANY OF US USE A MOBILE PHONE?**

AMONG CONSUMERS AGES 16+

86% - AUSTRALIA
84% - BRAZIL
89% - CHINA
81% - INDIA
97% - ITALY
98% - RUSSIA
99% - SOUTH KOREA
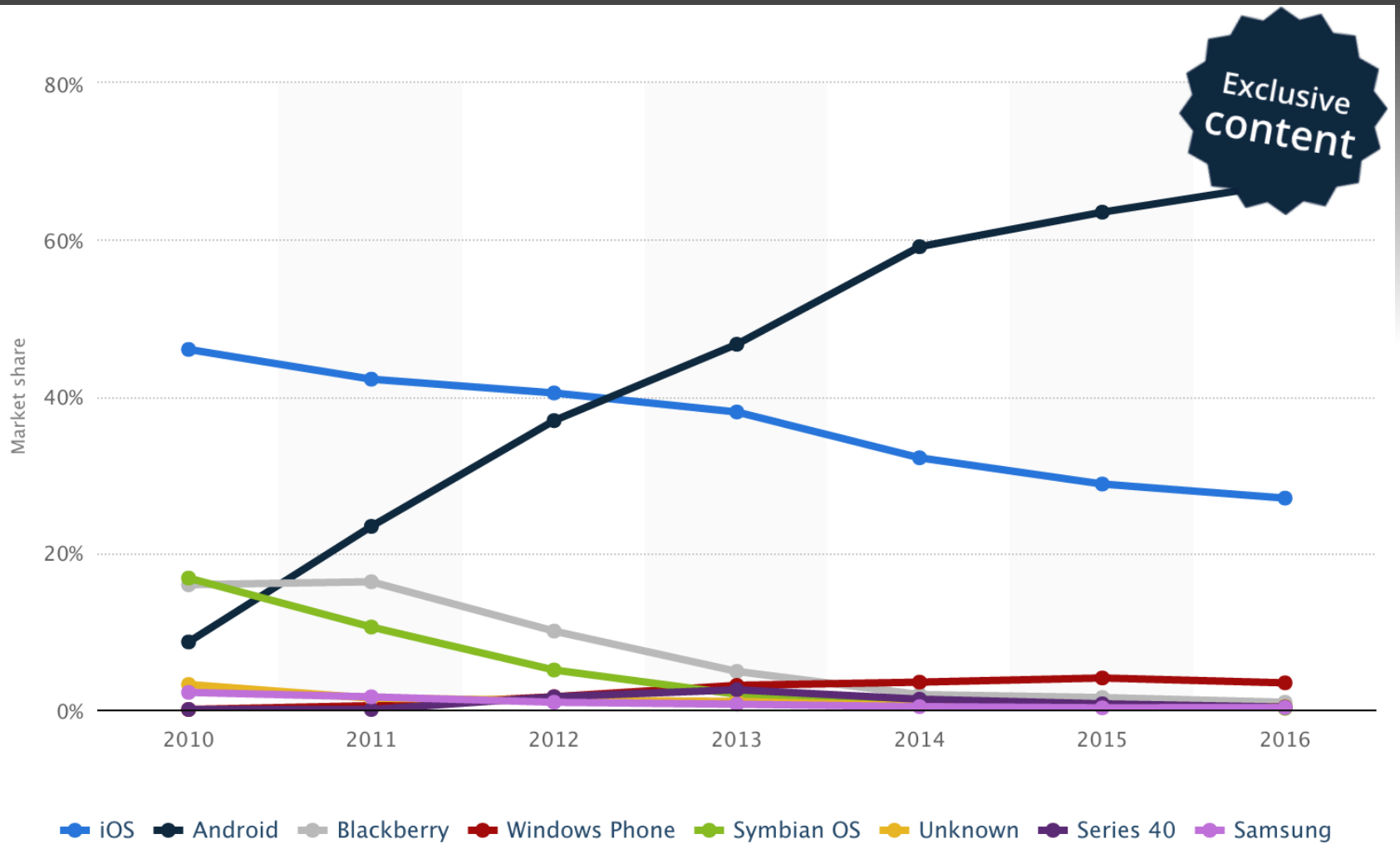97% - UNITED KINGDOM
94% - UNITED STATES
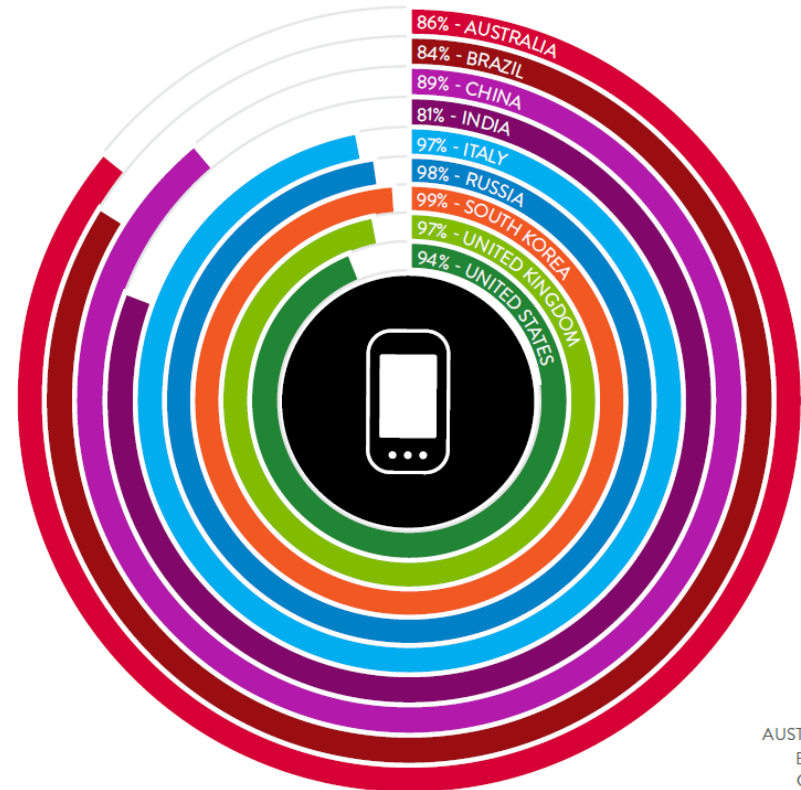
AUSTRALIA ●
BRAZIL ●
CHINA ●
INDIA ●
ITALY ●
RUSSIA ●
SOUTH KOREA ●
UNITED KINGDOM ●
UNITED STATES ●

Sources: Nielsen Global Smartphone Insights, first half of 2012; Nielsen Mobile Insights, 2012

TU/e
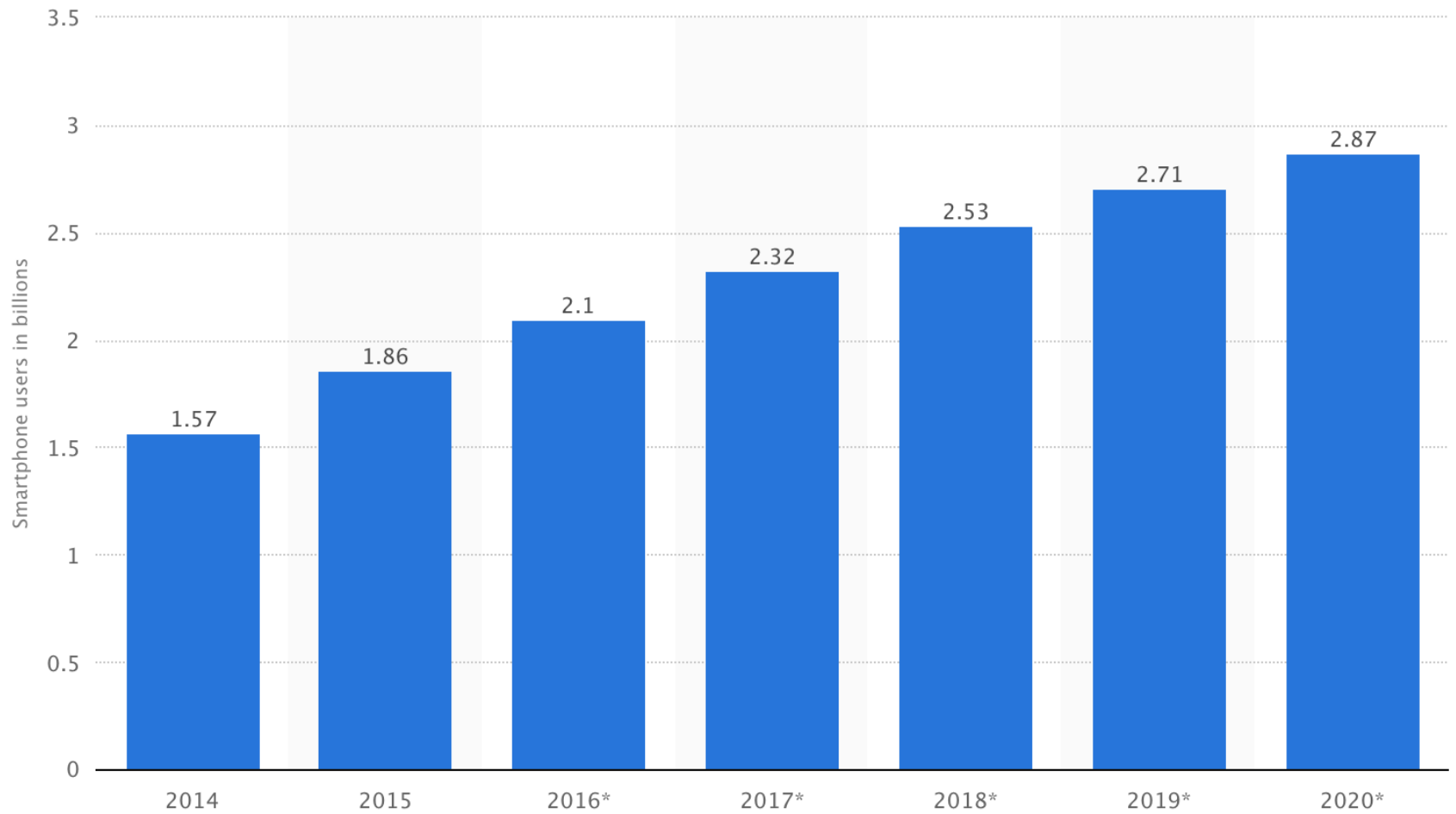
Smartphone Users - Globally (© Statista 2016)

Market share of mobile OSs in Europe, 2010 to 2016 (© Statista 2016)

TU/e

# LECTURE'S OBJECTIVES

1. Introduce app design guidelines
2. Raise awareness about mobile development technologies
3. Demo of INTEL XDK

TU/e

# OBJECTIVE (1ˢᵗ)

- Introduce app design guidelines

TU/e

# Question

- If you had to choose one aspect of mobile interaction that is crucial to design, what would that be?
- Take a minute to think about it and write it

TU/e

# Challenges for mobile interaction?

TU/e

# Challenges for mobile interaction

- Requires visual attention
- Context dependent
- Limited screen
- Limited input
- Processor
- Energy

TU/e

# Opportunities?

TU/e

# Opportunities - mobile interaction

- Personal
- Sensors
  - Microphone
  - Camera
  - Accelerometer
  - Light
  - GPS
  - RFID/NFC
  - Compass
  - Gyroscope
  - Proximity

- (almost) always with user
- Connected
  - 4G, WiFi, Bluetooth
- Touch, gestures (NUI)

CLI • Codified • Strict → GUI • Metaphor • Exploratory → NUI • Direct • Intuitive
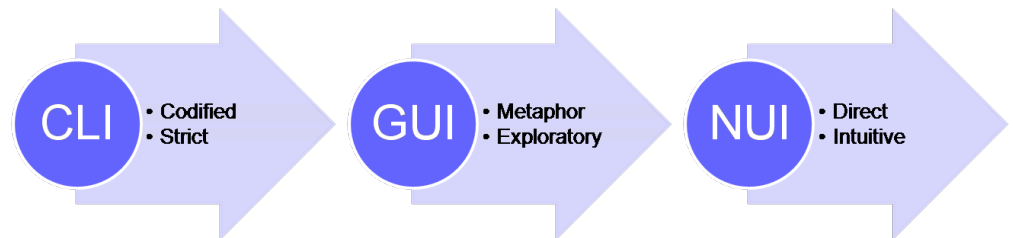
By August de los Reyes - Re-created, Public Domain, https://commons.wikimedia.org/w/index.php?curid=41476193

TU/e

# General to specific guidelines

TU/e

# Technology Acceptance Model (TAM)

- **Perceived usefulness**
  - is defined as the extent to which a person believes that using the system will enhance his or her job performance
- **Perceived ease of use**
  - is defined as the extent to which a person believes that using the system will be free of effort.

Davis, F. D. (1989), "Perceived usefulness, perceived ease of use, and user acceptance of information technology", MIS Quarterly 13(3): 319–340

**TU/e**

# TAM: Usefulness

- Extending functionality
  - Enlarging screen estate for collaboration
  - Extending keyboard for typing
  - APIs

TU/e

# Andres Lucero (USI2002)

- ## Pass-them-around
- [Lucero, A., Holopainen, J., & Jokela, T. (2011, May). Pass-them-around: collaborative use of mobile phones for photo sharing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1787-1796). ACM.]

- https://youtu.be/7JY7ZCTvmDA

TU/e

# Sandwich Keyboard: Fast Ten-Finger Typing on a Mobile Device with Adaptive Touch Sensing on the Back Side

**Oliver Schoenleben**
Helsinki Institute for Information Technology
HIIT, Aalto University and University of Helsinki
Otaniementie 19B, 02150 Espoo, Finland

**Antti Oulasvirta**
Max Planck Institute for Informatics
Saarland University, Campus E1 7,
66123 Saarbrücken, Germany
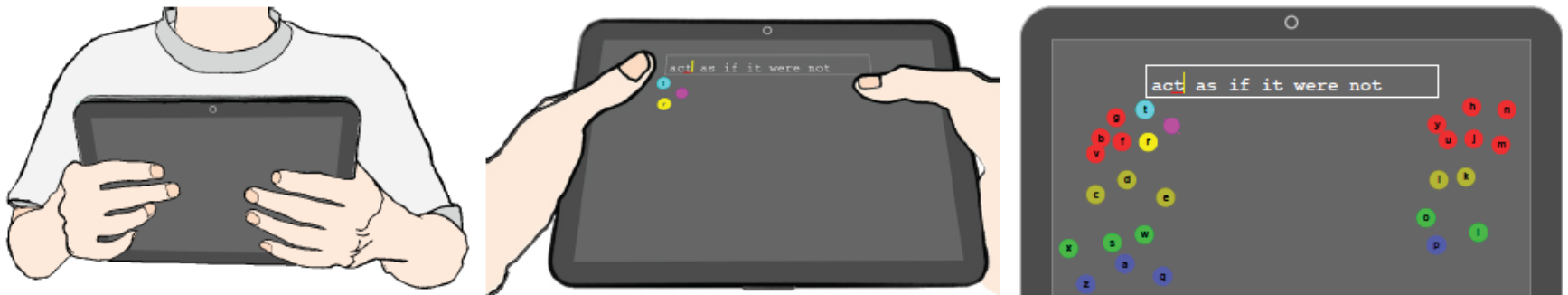
Figure 1. Sandwich Keyboard is gripped from its sides such that the thumbs type on the front side (middle) and the other fingers on the back side (left). Key targets of the keyboard in the back can be projected to the front to aid in learning the technique (right). Colors denote finger assignment.

TU/e

# APIs

- List of APIs
  - >13K @ [programmableweb.com](programmableweb.com)
  - [https://www.mashape.com/explore](https://www.mashape.com/explore)
- Crowd APIs
  - [http://sandbox.gengo.com](http://sandbox.gengo.com)
  - [Amazon Mechanical Turk API Reference](Amazon Mechanical Turk API Reference)
  - [Microworkers API tutorial](Microworkers API tutorial)

TU/e

# TAM: Ease of Use

TU/e

# Industry perspective

TU/e

# CONTEXT IS KING
[Gualtieri, M. (2011). Mobile App Design Best Practices. Forrester Research]

**Figure 5** The Five Dimensions Of Mobile UX Context: LLIID

| | |
|---|---|
| **Location** | Mobile users can use their mobile device wherever they are. |
| **Locomotion** | Mobile users can use their mobile device on the go. Users may be walking, running, in a car, on a boat, or in any other manner of motion while using a mobile device. |
| **Immediacy** | Mobile users can use their mobile device at a moment's notice. |
| **Intimacy** | Mobile users may use multiple devices for different purposes, and a device's use can vary from a digital appendage to an occasionally used device for a specific personal or work task. |
| **Device** | • Mobile devices vary greatly in form factor and capabilities.<br>• Smartphones, eReaders, tablets, and other devices. |

59132　　　　　　　　　　　　　　　　　　　　Source: Forrester Research, Inc.

TU/e

# Apple's Human Interface Guidelines (HIG)

https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html

**TU/e**

# Starting and Stopping

- Start Instantly, avoid displaying a splash screen
- Avoid asking people to supply setup information
  - Set up for the needs of 80% of your users
  - Get as much info as possible from other sources
  - If you must ask for setup information, prompt people to enter it within your app
- Delay a login requirement for as long as possible
  - App Store doesn't ask login until buying something
  - If necessary provide a brief, friendly explanation with the reasons and how it benefits users

TU/e

# Starting and Stopping

- **Think carefully before providing** *onboarding*
- **If necessary:**
  - Give only the info to get started
  - Use animation and interactivity to engage users and help them learn by doing
  - Make it easy to dismiss or skip the onboarding experience

- **Avoid asking users to rate app too soon**

**TU/e**
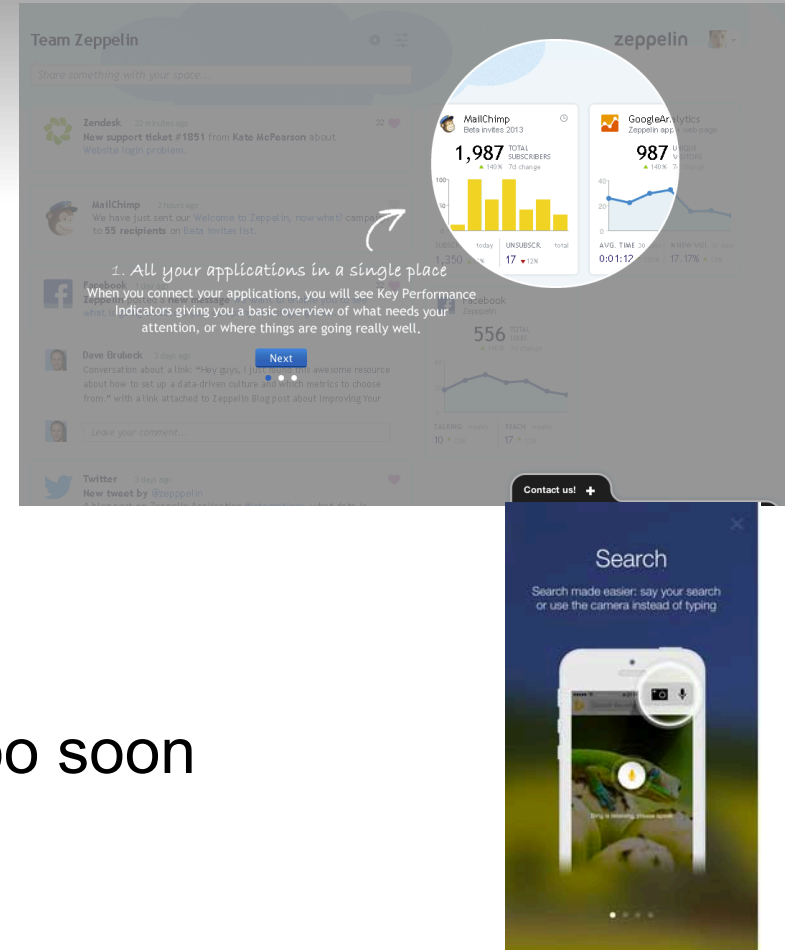
# Starting and Stopping

- In general, launch in the device's current orientation
- If app runs in only one orientation, you should always launch in that orientation

TU/e
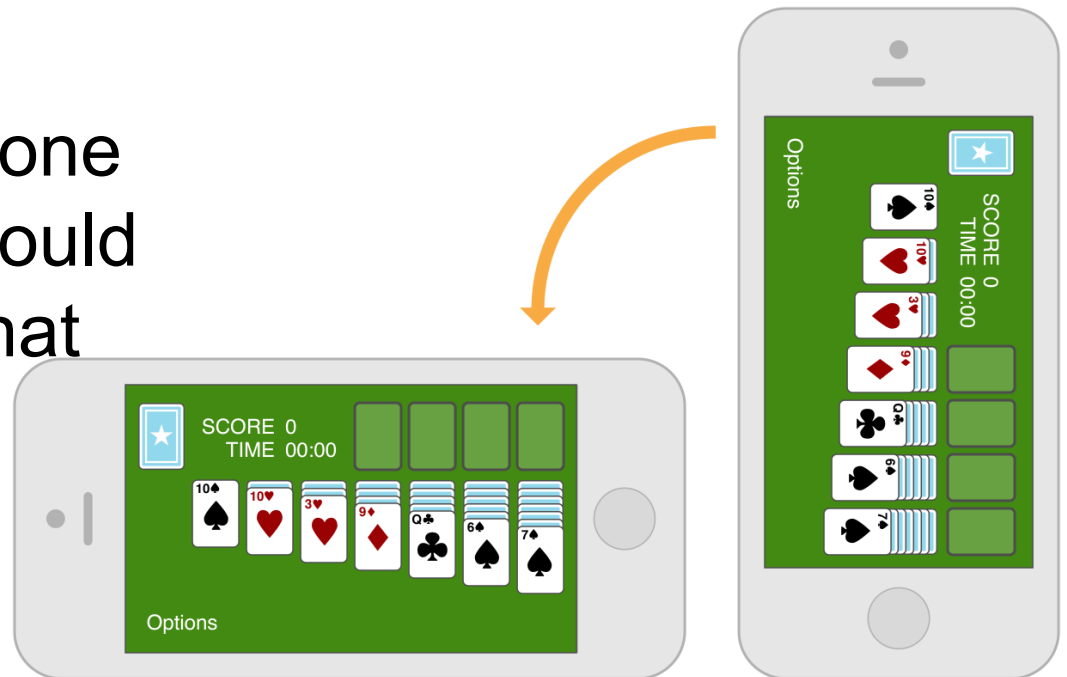
# Starting and Stopping

- Always Be Prepared to Stop
  - Save user data as soon as possible and as often as reasonable
  - When app restarts, restore its state

TU/e

# Basic guidelines

- Deference
  - The UI helps people understand and interact with the content, but never competes with it
- Clarity
  - Text is legible at every size, icons are precise and lucid, adornments are subtle and appropriate, and a sharpened focus on functionality motivates the design
- Depth
  - Visual layers and realistic motion impart vitality and heighten people's delight and understanding

TU/e

# Defer to Content (content is king)

- What is the core functionality?
- Although crisp, beautiful UI and fluid motion are highlights, the content is at its heart
- Guidelines:
  - Take advantage of the whole screen
  - Reconsider visual indicators of physicality and realism

TU/e
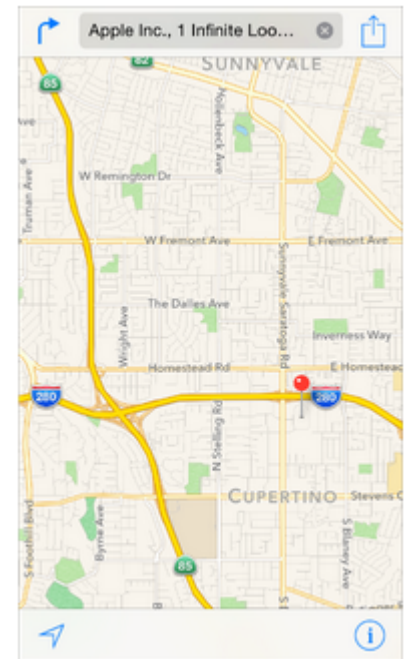
# Provide Clarity

- Clarity is another way to ensure that content is king
- Guidelines:
  - Use plenty of negative space
  - Let color simplify the UI
  - Embrace borderless buttons

**TU/e**
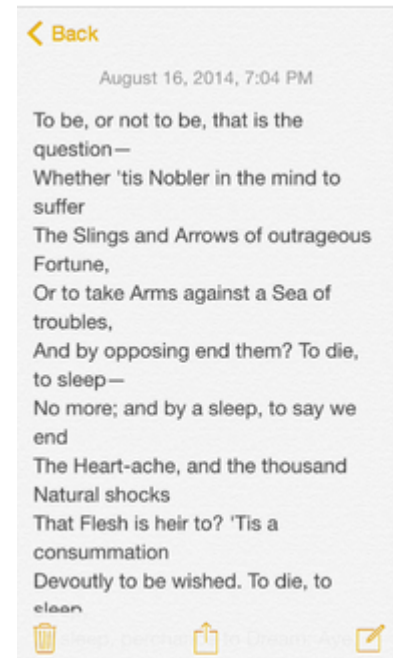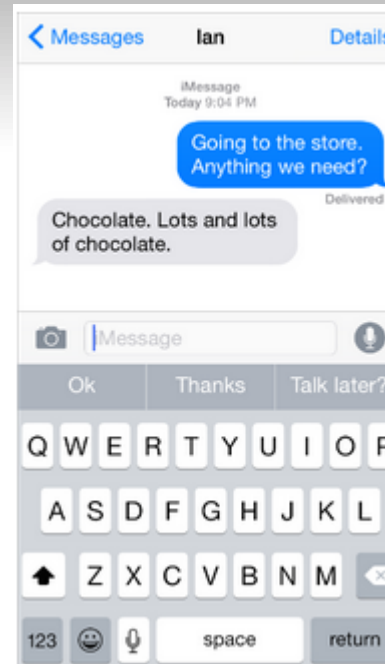
- Give each interactive element ample spacing (~7mmx7mm)



44 x 44

TU/e
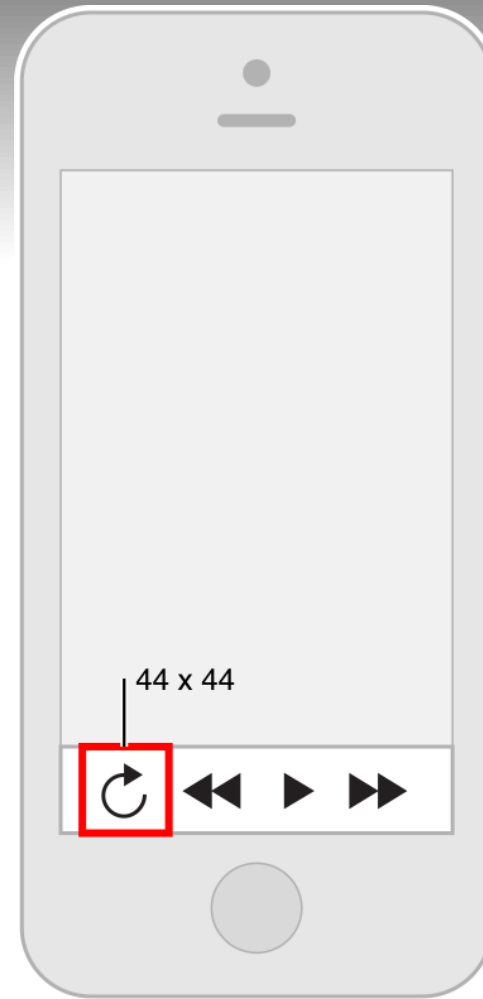
# Use Depth to Communicate

- Guidelines:
  - Translucent background appearing to float above the screen
  - List in layers. When users work with one list, the other lists are collected together at the bottom of the screen
  - Use transitions to give a sense of depth
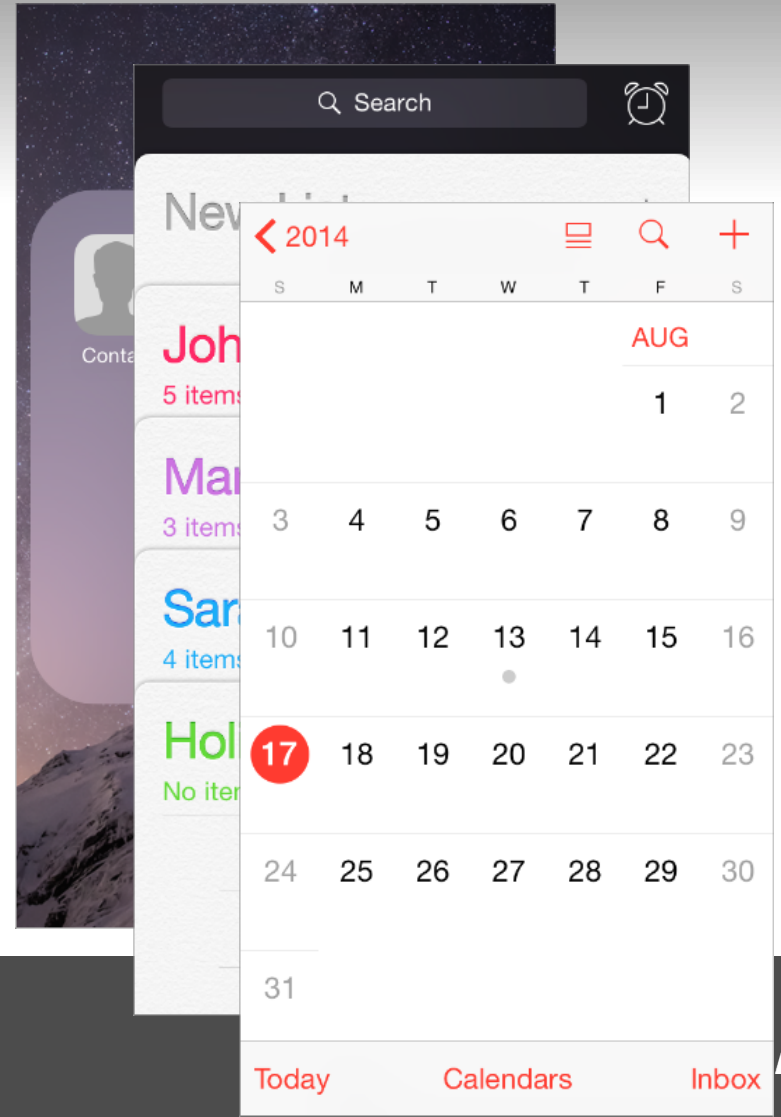
# Adaptivity and Layout

- **Guidelines**
  - Maintain focus on the primary content in all orientations
  - Avoid unjustifiable changes in layout
  - Users expect apps in different orientations

Regular height

Compact width

Compact height

Regular width

TU/e
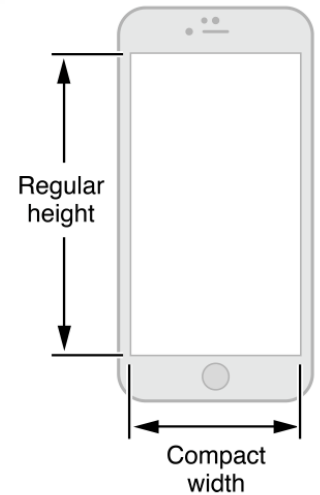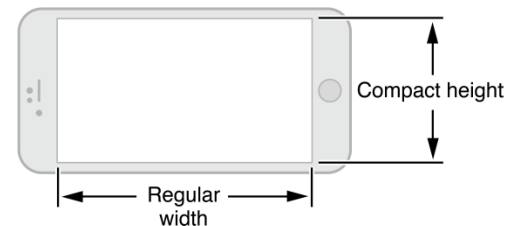
# Use Layout to Communicate

- Place principal items in the upper half of the screen
- Near the left side of the screen (in left-to-right cultures)

More important

Less important

TU/e
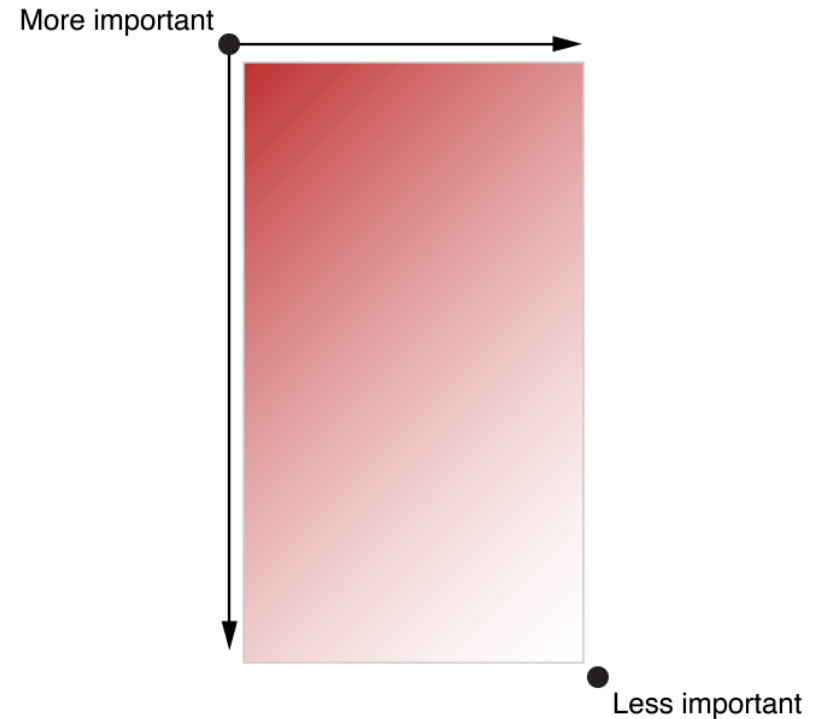
# Visual "weight"

- Use visual weight to show the relative importance of onscreen elements
- Large items catch the eye and appear more important
- Larger items are easier to tap, especially useful in apps that <u>users often use in distracting surroundings</u>
- Use indentation and alignment to ease scanning and communicate groupings

TU/e
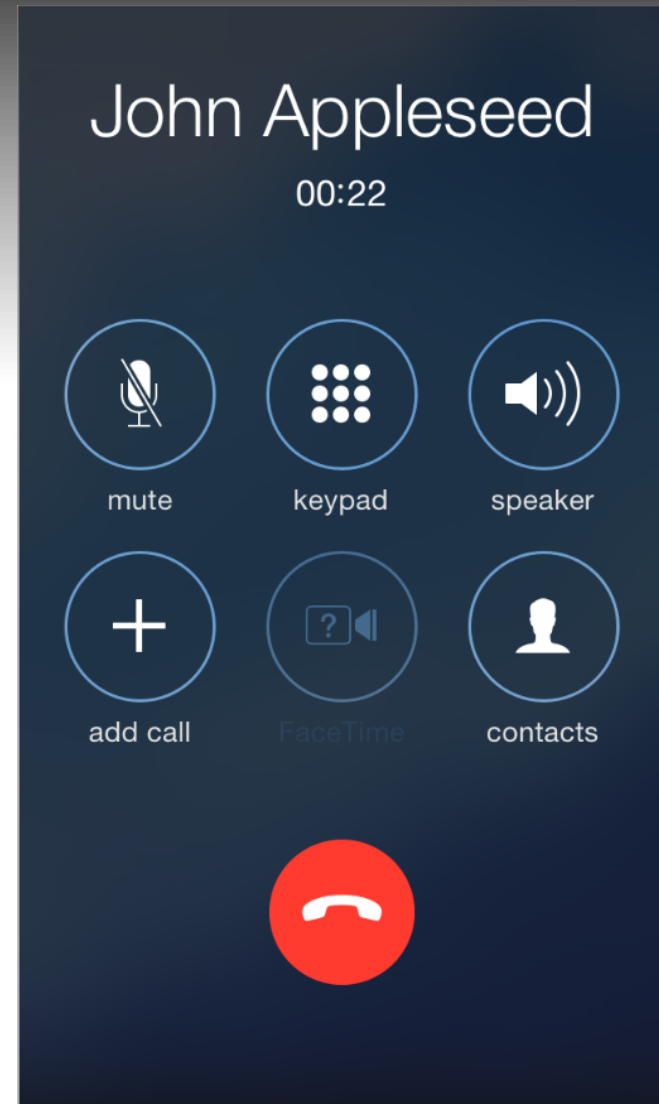
# Inputting Information Should Be Easy

- **Make it easy for users to make choices**
  - List instead of text
- **Get information from device, when appropriate**
- **Balance a request for input by giving users something useful in return**

TU/e
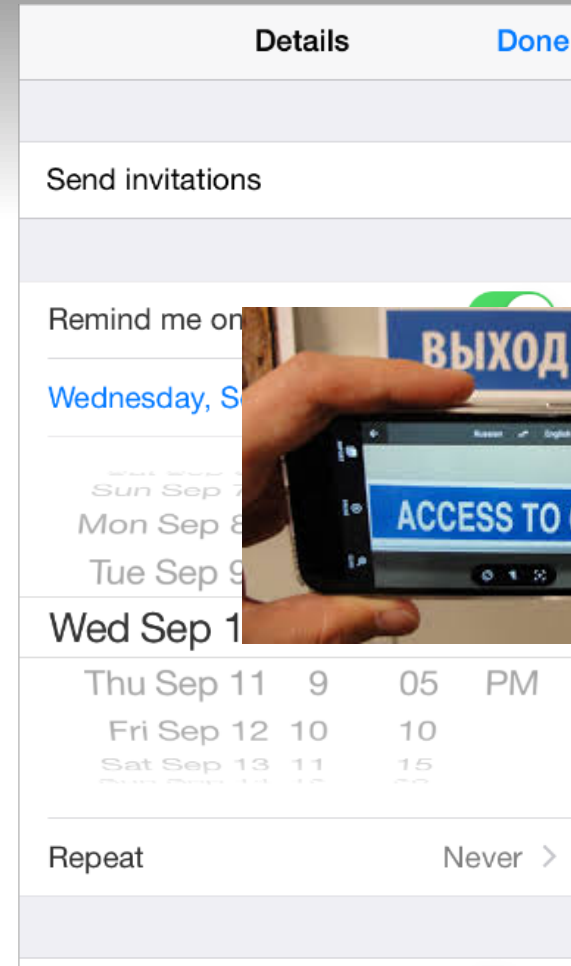
# Design Principles

- **Aesthetic Integrity**
  - Not a measure of the an app's beauty of artwork
  - Rather, it represents how well an app's appearance and behavior integrates with its function to send a coherent message

TU/e

# Design Principles

- Consistency
  - Consistency lets users **transfer their knowledge and skills** from one part of **an app's** UI to another and from **one app to another app**
  - Consistency does not mean verbatim copy of other apps; rather, it pays attention to the standards and paradigms people are comfortable with

TU/e

# External consistency of gestures


Tap
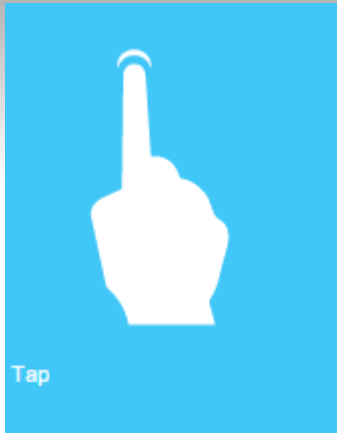
- To press or select a control or item


Touch and Hold
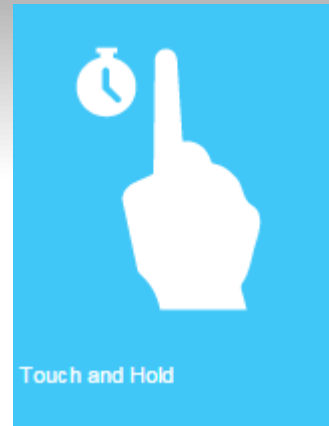
- In editable or selectable text, to display a magnified view for cursor positioning


Double Tap
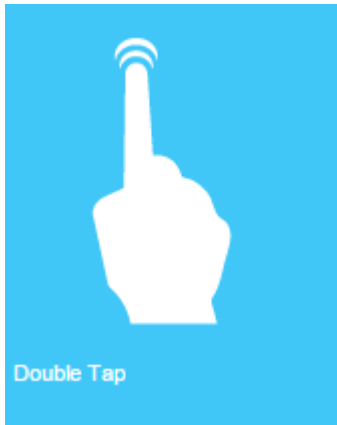
- To zoom in and center a block of content or an image


Pinch

- To zoom in

Icons from: http://gesturecons.com/

TU/e
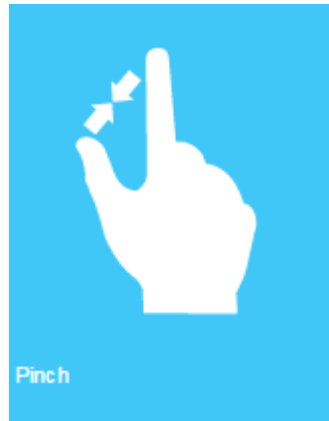
# Design Principles

- **Direct Manipulation**

  - When people directly manipulate onscreen objects instead of using separate controls to manipulate them, they're more engaged with their task and it's easier for them to understand the results of their actions

TU/e
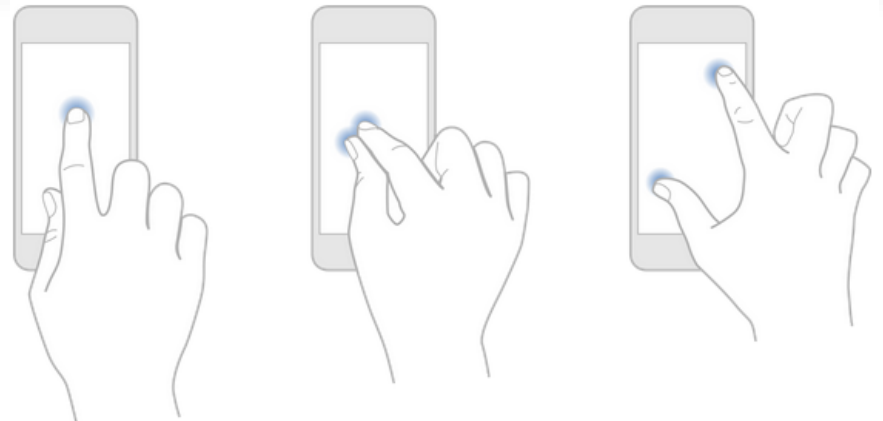
# OBJECTIVE (2nd)

- Raise awareness about mobile development technologies

TU/e

# AV (APP VITAE)

TU/e

# AV (APP VITAE)

**TU/e**

# AV (APP VITAE)

TU/e

# AV (APP VITAE)

TU/e

# MOBDEV TECH

TU/e

# MOBDEV TECH

TU/e

# MOBDEV TECH

| | **Native** | **Using Web tech (Hybrid)** | **WYSIWYG** |
|---|---|---|---|
| Technical skills | ooooo | oo | o |
| Native UI elements | ✔ | - | - |
| Ability to integrate phone sensors | ✔ | ✔ | - |
| Cross-platform | - | ✔ | ✔ |
| Flexibility for developing features | ✔ | ✔ | - |

TU/e

# ADVANTAGES

- Cross-platform
- Unified UI
- Simplified development
- Progressive enhancement
- Developing not just an app but also learning how to develop a mobile-friendly website

TU/e

# WORKING ENVIRONMENT
## (old −but still valid!)

- **For Editing**


Brackets

- **For User Interface**


jQuery mobile framework.

- **For Packaging into an app**
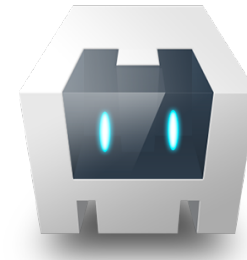


- Great tools for Rapid Prototyping
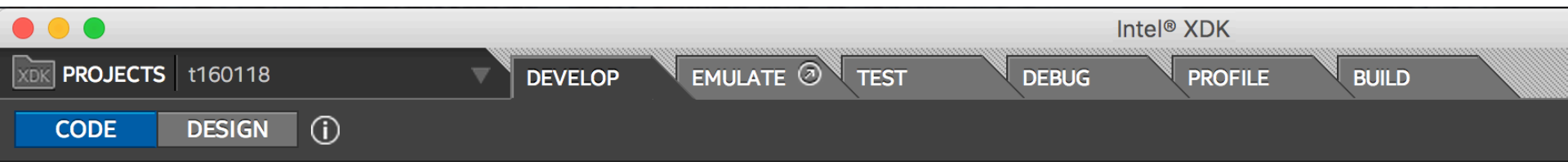  - (not necessarily for performance)
  - More tools are appearing as we speak

TU/e

# WORKING ENVIRONMENT

- INTEL XDK is unifying these (IDE)
- "*The Intel XDK application consists of a set of development tools to help you code, debug, test, and build mobile web apps and hybrid HTML5 apps for multiple target platforms.* "
  [https://software.intel.com/en-us/xdk/docs/intel-xdk-introduction]



- +App preview on your phone:

TU/e

# CLIENT-SIDE SCRIPTING
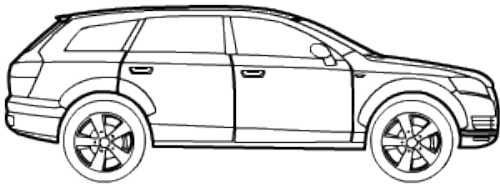


Structure

Style

Behavior

TU/e

# WHAT IS JAVASCRIPT?

- JavaScript is a scripting language mainly used for creating dynamic websites
- JavaScript was initially developed by Netscape

- It is NOT Java!

TU/e

# WHAT IS JQUERY?

- [jQuery](#) is a fast and concise JavaScript library
- Practically speaking it is helping you to write JavaScript more efficiently

TU/e

# WHAT IS JQUERY MOBILE?

- jQuery Mobile (jQM): Touch-Optimized Web Framework for Smartphones & Tablets
- Developed on HTML5, CSS and jQuery

- With jQM you essentially build cross-browser, mobile-friendly websites

TU/e

# WHAT IS CORDOVA?

- Cordova is an open source framework for quickly building cross-platform mobile apps using HTML5, Javascript and CSS https://cordova.apache.org/

TU/e
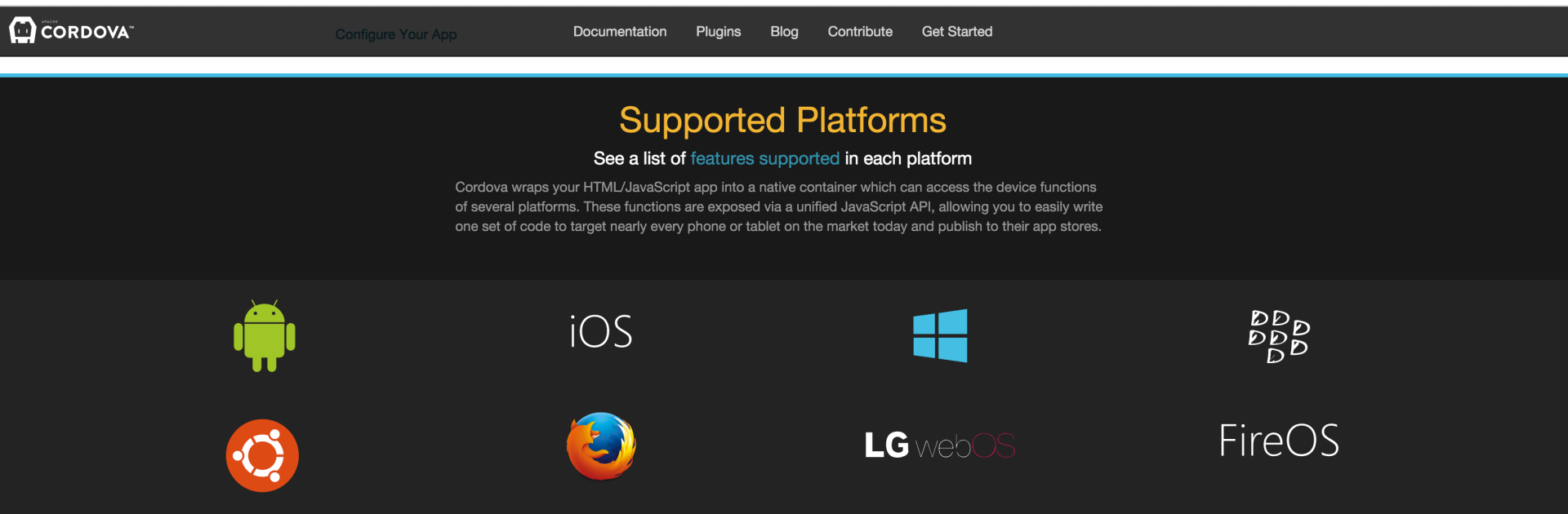
# WEB APPS vs. HYBRID APPS

- Rely on web server and accessed via the Internet
- Run in a mobile browser and originate from and have access to a network-addressable server

http://blogs.intel.com/evangelists/2015/12/31/five-useful-tips-on-getting-started-building-cordova-mobile-apps-with-the-intel-xdk/

- Have no server and no guarantee that they will ever have a network connection!
- Run within a webview,
- Are simply files stored in a *package* on the mobile device's filesystem (i.e. not complied)

TU/e

# HOW DOES IT WORK? (SYSTEM LEVEL - MACRO)

- Cordova has pre baked native apps for different OSs that wrap a *webview*
- Whatever HTML5 code you write is interpreted within that *webview*
- Access to mobile sensors (camera, mic, GPS, etc.) & mobile information (contacts, battery status & other device info)

TU/e

# ARCHITECTURE - MACRO

- Cordova wraps a webview

- Developer writes HTML, CSS, JS

- Cordova interfaces sensors with webview and packages the HTML, CSS, JS files

TU/e

# ARCHITECTURE - MESO

- "Multi-page, single file" architecture
  - i.e. many pages in one html file

- "Multi-page, multi file" architecture
  - i.e. many pages in many html files

TU/e

# HOW DOES IT WORK? (PROGRAMING LEVEL - MICRO)

- JQM utilizes custom *data-* attributes and has special classes build-in

TU/e

# HTML5 - STRUCTURE

- Elements have attributes
  - *html* element has the attribute *lang*
  - the attribute *lang* has *value* en

```
<html lang="en">
<head>
<title>Swapping Songs</title>
</head>
<body>
<h1>Swapping Songs</h1>
<p>Tonight I swapped some of the songs I wrote with some friends, who
gave me some of the songs they wrote. I love sharing my music.</p>
</body>
</html>
```

TU/e

# LISTVIEW

- **Count bubble**
  - Add a *<span>* with *class="ui-li-count"* to a listview item (*<li>*)
  - E.g. *<span class="ui-li-count">2</span>*

```
<li data-theme="c">
    <a href="#page3" data-transition="slide">
        Lecturers
        <span class="ui-li-count">
            2
        </span>
    </a>
</li>
```

TU/e

# LISTVIEW

- Change icon
  - Add to <li> attribute *data-icon*

```
<li data-theme="c" data-icon="info">
    <a href="#page5" data-transition="slide">
        About
    </a>
</li>
```

About                                              ⓘ

TU/e

# LISTVIEW

- **Adding icons**
  - Add an *<img>* inside an <a> of a listview item
  - Add the *class="ui-li-icon"* to the *<img>*

**TU/e**
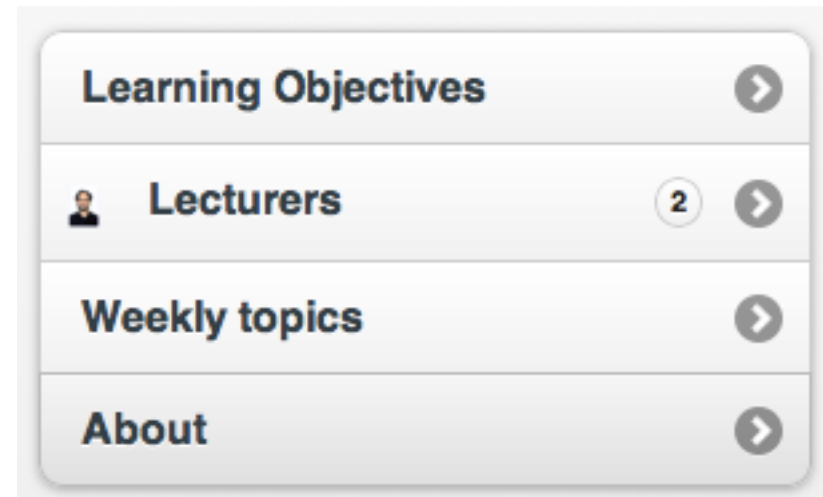
# MOST UI ELEMENTS

- **Change themes**
  - Add attribute: *data-theme* to any element
  - Choose a value: *a, b, c, d, e, f*

```
<div data-role="page" id="page">
    <div data-role="header" data-theme="">
        <h1>SP1-02</h1>
    </div>
    <div data-role="content">
        <ul data-role="listview">
            <li><a href="#page2">Learnin
            <li><a href="#page3">Lecture
            <li><a href="#page4">Weekly Schedule</a><span class=
"ui-li-count">7</span></li>
        </ul>
```

a
b
c
d
e
f

SP1-02

Learning Objectives

Lecturers

Weekly Schedule

NHTV

```
<div data-role="page" id="page">
    <div data-role="header" data-theme="b">
        <h1>SP1-02</h1>
    </div>
```

SP1-02

Learning Objectives

TU/e
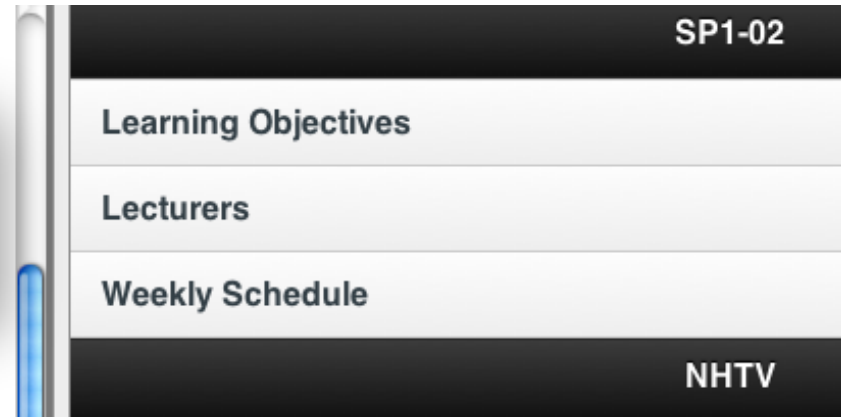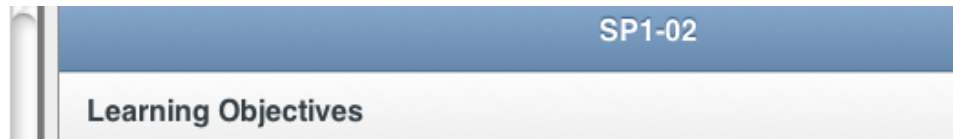
# BASIC TEMPLATE PAGE

```html
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title></title>
6          <link rel="stylesheet" href="jquery.mobile-1.2.0.min.css" />
7          <script src="jquery-1.7.2.min.js"></script>
8          <script src="jquery.mobile-1.2.0.min.js"></script>
9      </head>
10
11     <body>
12       <div data-role="page">
13           <div data-role="header" data-theme="b">
14               <h3></h3>
15           </div>
16
17           <div data-role="content">
18
19           </div>
20
21           <div data-role="footer" data-theme="b">
22               <h3></h3>
23           </div>
24       </div>
25     </body>
26  </html>
27
```

- For linking to CDN-hosted files: [http://jquerymobile.com/download/](http://jquerymobile.com/download/)

TU/e

# NOTE ON: EVENT-DRIVEN MODEL

- In computer programming, event-driven programming is a programming paradigm in which the flow of the program is determined by events such as user actions (mouse clicks, key presses), sensor outputs, or messages from other programs/threads.

- https://en.wikipedia.org/wiki/Event-driven_programming

- E.g.: http://www.w3schools.com/js/js_events.asp

```
<!DOCTYPE html>
<html>
<body>

<button
onclick="document.getElementById('demo')
.innerHTML=Date()">The time is?</button>

<p id="demo"></p>

</body>
</html>
```

The time is?

Sun Dec 11 2016 20:58:12 GMT+0100 (CET)

TU/e

# RESOURCES

- http://jquerymobile.com/
- https://cordova.apache.org/ - http://phonegap.com/
- https://cordova.apache.org/docs/en/latest/cordova/plugins/pluginapis.html


- Series of videos (screencasts) on jQM: http://www.youtube.com/playlist?list=PLBNlfBYNGprg2qIEdkoF5HTBp2UcvlM5P

TU/e

# OTHER INTERESTING TOOLS (for rapid prototyping)

- http://www.appinventor.org/
- http://ai2.appinventor.mit.edu/
- http://ionicframework.com/
- http://framework7.io/ (for UI)

TU/e

# YOU NEED:

- Preferably phone to test sensors that are hard to emulate

- To build you can also use: https://build.phonegap.com

TU/e

# Demo INTEL XDK

TU/e

# Lots of examples: start with

TU/e

# Documentation

- https://software.intel.com/en-us/xdk/docs/templates-to-get-started-with-xdk-app-development

TU/e

# Next

- Lots of tools out there that help you build an interactive prototype of your app
- Try: https://marvelapp.com/ or https://www.invisionapp.com/
- Have you found other tools? Please share!

- Next week: sensors

TU/e

# SUMMARY

- App design guidelines
- Awareness about mobile development technologies
- Workflow of Cordova / jQM

CONTACT      v.j.khan@tue.nl      khan.gr      @v_j_khan

TU/e