

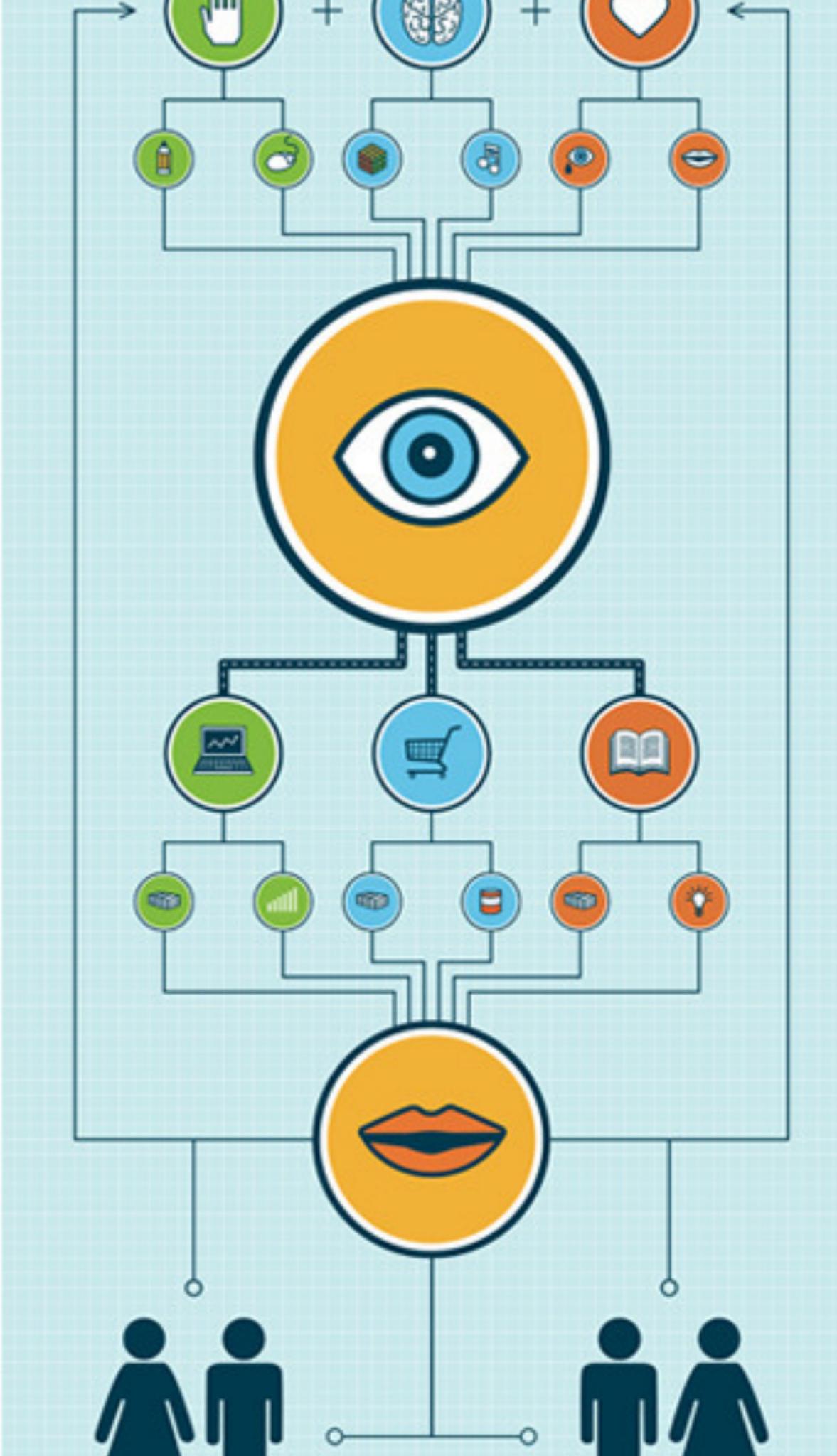
Mobile App Development 1

Human Interface Design

Robert O'Connor
roconnor@wit.ie 

Outline

- Human Interface Design
 - Design With The User In Mind
 - Mobile Design Principles
 - Practicalities





Why Is Good Design Important? - <http://www.youtube.com/watch?v=BuP29Zmglks>

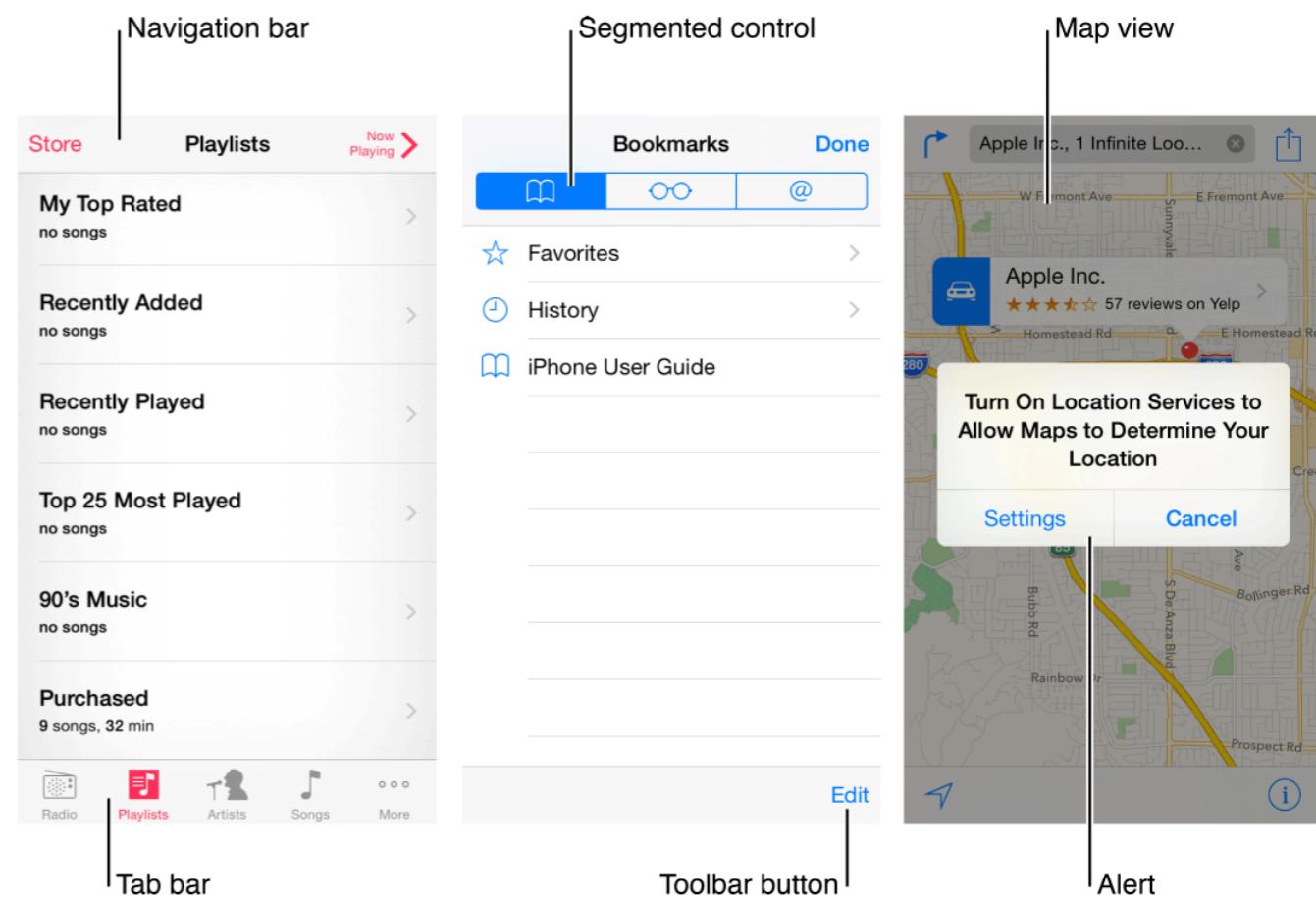
Design With The User In Mind

- The success of an app depends largely on the quality of its user interface. If users don't find an app attractive and easy to use, even the fastest, most powerful, most full-featured app can languish in the App Store.
- There are many ways to get from initial inspiration to popular app, and there is no single path that guarantees success.
- But there is one directive on which all successful app development depends: **Design with the user in mind.**



Design With The User In Mind

- Apple's **iOS Human Interface Guidelines** and Google's **Android Design** are very detailed documents describing strategies and principles surrounding smartphone app design
- <https://developer.apple.com/ios/human-interface-guidelines/>
- <https://developer.android.com/design/index.html>
- **YOU SHOULD READ THESE**



Understand How People Use Their Devices

- Consider how the following device and software features affect the user experience:
- Smartphones and tablets are handheld devices that enable and encourage people to use them on the go. People expect apps to start quickly and be easy to use in a wide variety of environments.
- The display is paramount, regardless of its size. The comparatively small device margin around the display becomes almost invisible while people are engaged in using apps.
- The Multi-Touch interface allows people to manipulate content without the intervention of another device, such as a mouse. People tend to feel more in control of the app experience because they can use touch to manipulate onscreen elements.

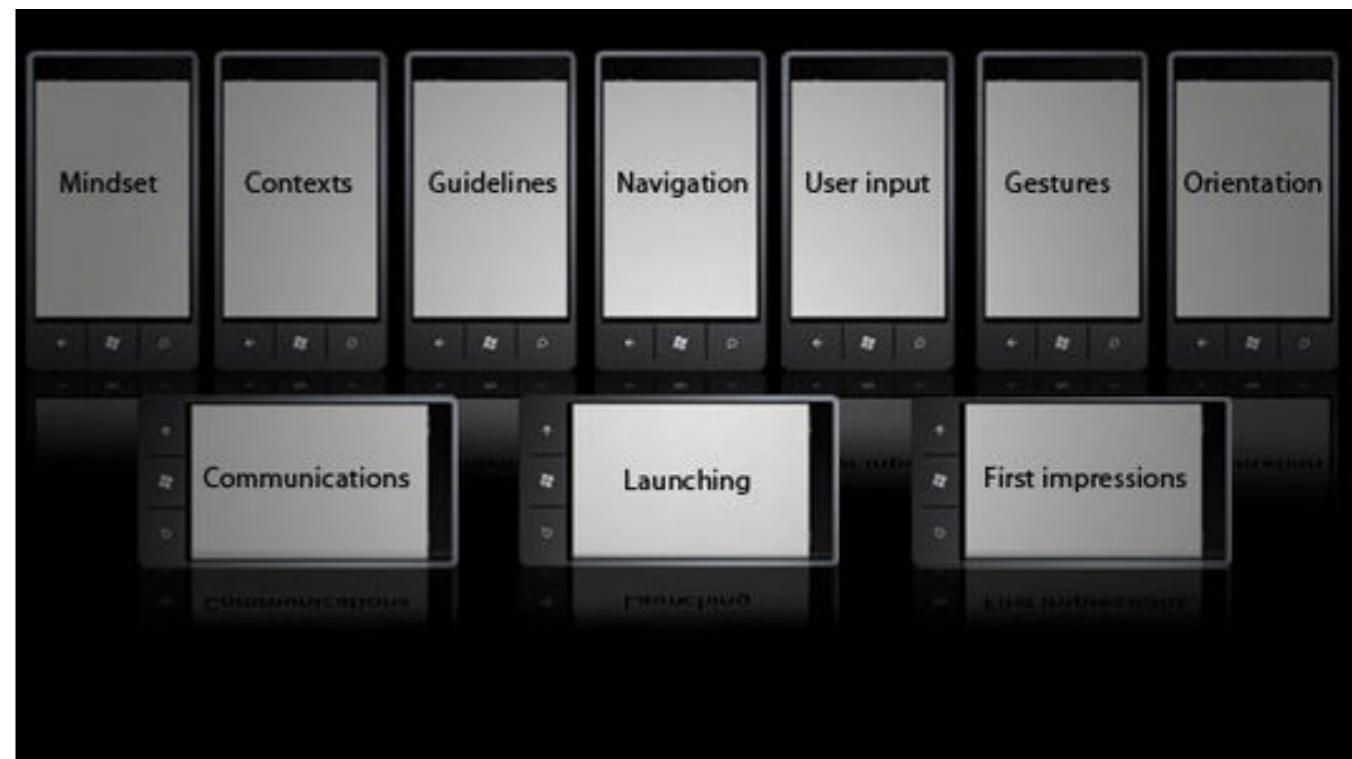
Understand How People Use Their Devices

- Only one app at a time is frontmost. Users can use the multitasking bar to switch between apps quickly and easily, but the experience differs from seeing multiple apps open simultaneously on a computer display.
- In general, apps don't open separate windows at the same time. Instead, users transition between screens of content, each of which can contain multiple views.



Mobile Design Principles

- Mobile software consultant **Jonathan Stark** published 10 Principles of Mobile Design on the popular Creative Bloq (Stark, 2012)
- Summarises many of the more *verbose* guidelines
- His author page on O'Reilly Media has some interesting webinars
- <http://www.oreilly.com/pub/au/3840>



Mobile Design Principles - Mobile Mindset

- **Be focused:** More is not better. Edit your features ruthlessly. You are **going to have to leave stuff out**



- **Be unique:** Know what makes your app different and amplify it. There are lots of fish in the sea of mobile apps. If there's nothing special about your app, why would anyone pick it?

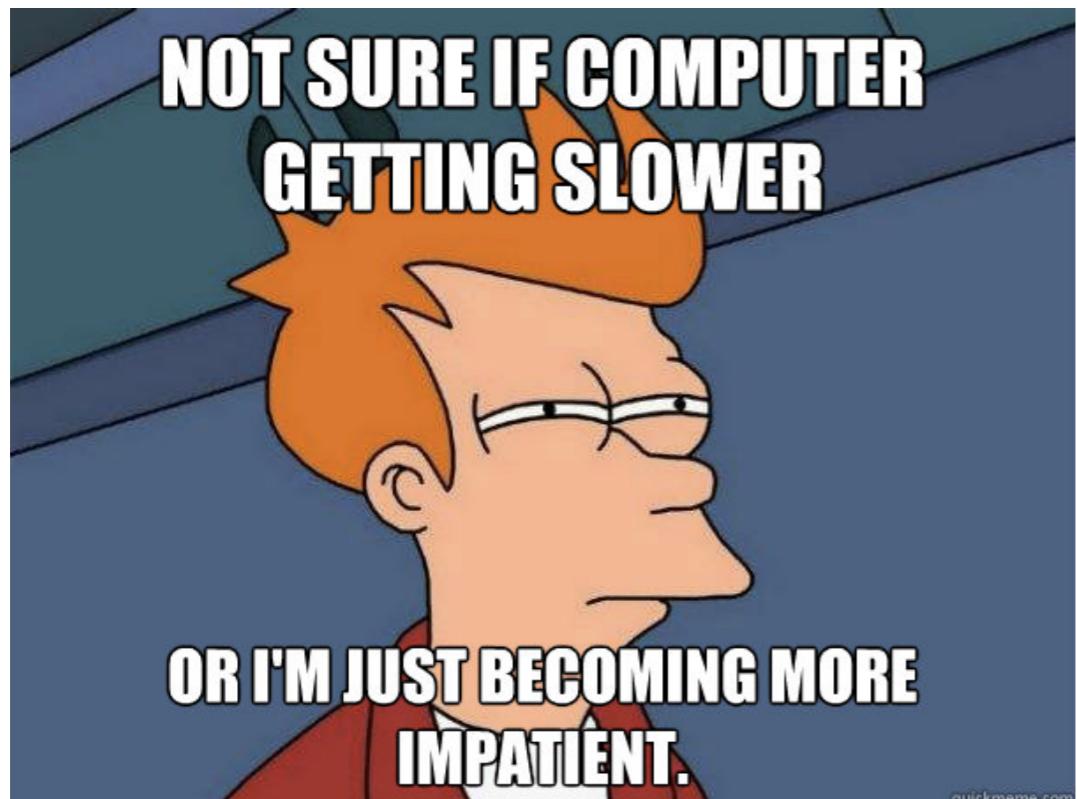
- **Be charming:** Mobile devices are intensely personal. They are our constant companions. Apps that are friendly, reliable and fun are a delight to use, and people will become quite attached to the experience.
- **Be considerate:** App developers too often focus on what would be fun to develop, their own mental model of the app or their personal business goals. These are good places to start, but you have to put yourself in your users' shoes if you ever hope to create an engaging experience.

Mobile Design Principles - Mobile Contexts

- Three major mobile contexts: Bored, Busy and Lost
- **Bored:** There are a lot of people using their smartphones on the couch at home. In this context, immersive and delightful experiences geared toward a longer usage session are a great fit. Still, interruptions are highly likely so be sure your app can pick up where your user left off. Examples: Facebook, Twitter, Angry Birds, web browser.
- **Busy:** This is the 'running through the airport' scenario. The ability to accomplish micro-tasks quickly and reliably with one hand in a hectic environment is critical. Remember that the user will have tunnel vision in this context, so huge targets and bold design are important. Examples: email, calendar, banking.
- **Lost:** Users who are in transit, in unfamiliar surroundings, or in familiar surroundings but interested in something unknown around fall into the lost category. In this context, sketchy connectivity and battery life are big concerns, so you should offer some level of offline support and be sparing with your use of geolocation and other battery hogs. Typical examples: Maps, Yelp, Foursquare.

Mobile Design Principles - Global Guidelines

- **Responsiveness:** Responsiveness is absolutely critical. If your user does something, your app needs to acknowledge the interaction instantly. Note that responsiveness and speed are not the same thing. It's OK if certain operations take time. Just make sure you let the user know you're working on it.



- **Polish:** Polish is extremely valuable. Because of the 'constant companion' nature of our relationship to smartphones, paying a lot of attention to getting the little details perfect will be noticed and appreciated. I think of this as being like the 'fit and finish' of a car. The engine might be powerful and the body style gorgeous, but if there's a lot of road noise or rattling on the highway, the experience will begin to degrade for the commuter.
- **Thumbs:** With the advent of touchscreen interfaces, everyone is always talking about 'finger this' and 'finger that'. In reality, the thumb is what we need to design for. Unless the user is interacting with her smartphone with two hands, it's almost impossible to get a finger on the screen. And even in a two handed grip, she's likely to type with two thumbs. Thumbs are the default.

Mobile Design Principles - Global Guidelines

- **Targets:** Look at your right thumb. Not the tip, but the face – the part that comes into contact with your phone screen. Mine is the approximate size and shape of a bottle cap. Great for a lot of things, but far from precise when it comes to targeting tiny regions of my smartphone.
- The magic number for thumb friendly UI elements is 44 pixels. Exceptions abound, but this is a good “rule of thumb.” You should also be conscious of where you place your targets relative to each other. For example, putting the Backspace button directly adjacent to the Send button in an SMS app would be a bad idea.



Mobile Design Principles

- Global Guidelines

- **Content:** The revolution of touch interfaces is that they enable us to interact directly with our content. This removes abstractions (such as mouse and trackpad) and is more in line with how our brains are wired. Leverage the intuitive power of touch UI by minimising interface chrome (buttons, tab bars, checkboxes, sliders and so on) wherever possible and putting your content front and centre.

Clean gutters

Wash car

Switch laundry

Submit article

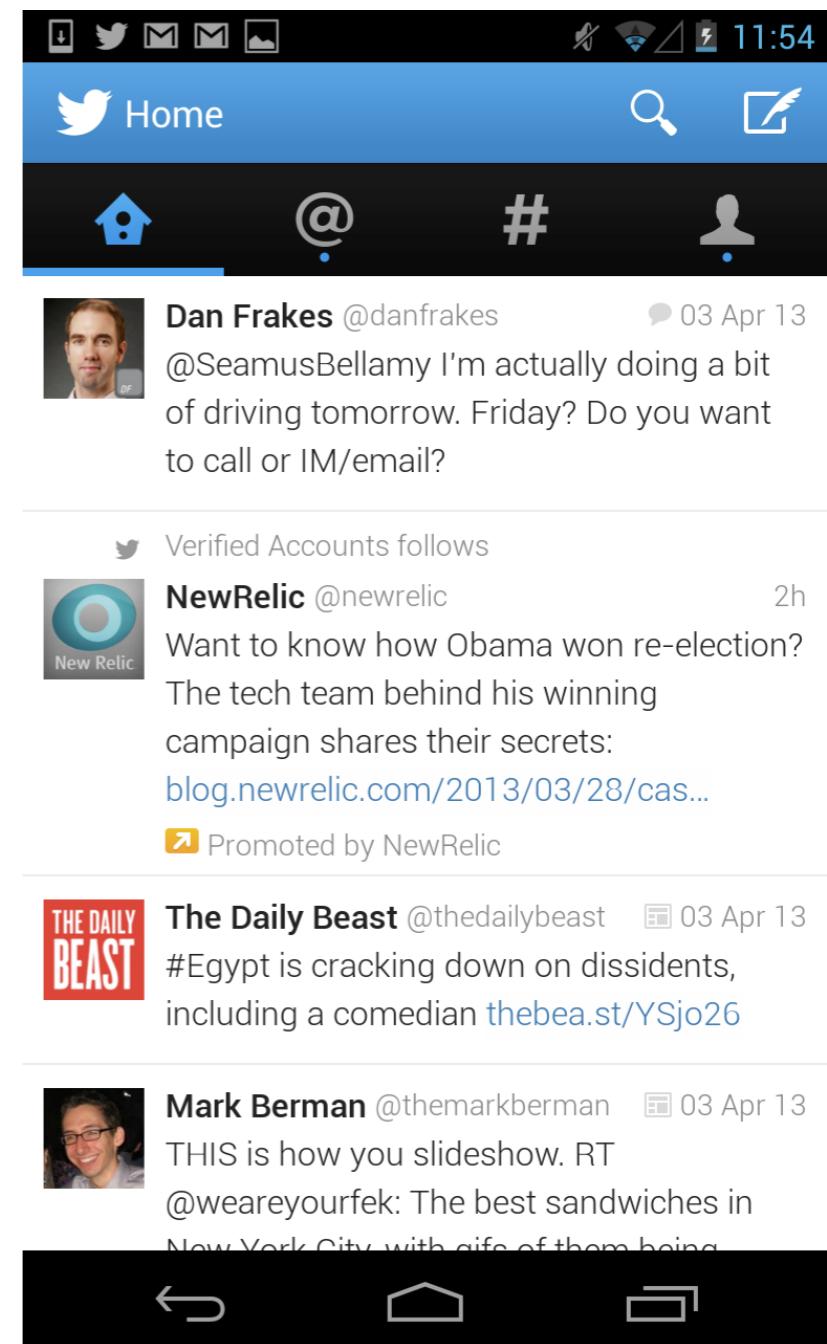
Backup computer

Mobile Design Principles - Global Guidelines

- **Controls:** When you do have to add controls, try to put them at the bottom of the screen (in other words beneath the content). Think of an adding machine, a bathroom scale or even a computer – the controls are beneath the display. If they weren't, we wouldn't be able to see what was going on with the content while we were using them.
- **Contrast** this real-world design consideration with traditional web or desktop software, where navigation and menu bars are virtually always at the top. This makes sense in a mouse context because the pointer is nearly invisible. Not so with the 'meat pointer' at the end of your arm.
- **Scrolling:** Avoid scrolling. I can assure you that 'below the fold' exists for mobile. Also, having a non-scrolling screen has a more solid and dependable 'feel' than a scrolling view because it's more predictable. Of course, certain screens have to scroll, but it's good to avoid it where you can. If you think discoverability might be an issue, you can reverse animate scrollable content into its default position to give a subtle but effective indication that there is more content out of view.

Mobile Design Principles - Navigation Models

- If you're going to use one of common navigation models, be sure to pick the one that makes the most sense for your app.
- **None:** Single screen utility apps (eg Weather app on iPhone)
- **Tab bar:** Three to six distinct content areas (eg Twitter for iPhone)
- **Drill down:** List and detail content hierarchy (eg Settings app on iPhone)



Mobile Design Principles - User input

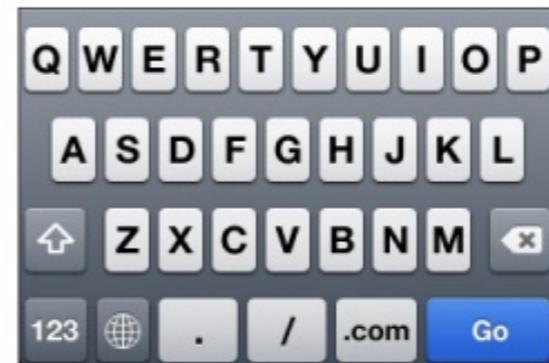
- Typing can be painful on smartphone devices
- There are about a dozen keyboard variations on popular smartphones (text, number, email, URL and so on). Consider each of your input fields and be sure to display the keyboard that will be most useful for the data entry being done.
- Auto-correct can be so hilariously frustrating that there is a website devoted to it. Consider each of your input fields and decide which auto entry options should be enabled (such as auto-correct, auto-capitalisation and auto-complete).
- If your app invites a lot of typing, you should ensure you support landscape orientation for fat-thumbed folks like me.



Default



Email



URL



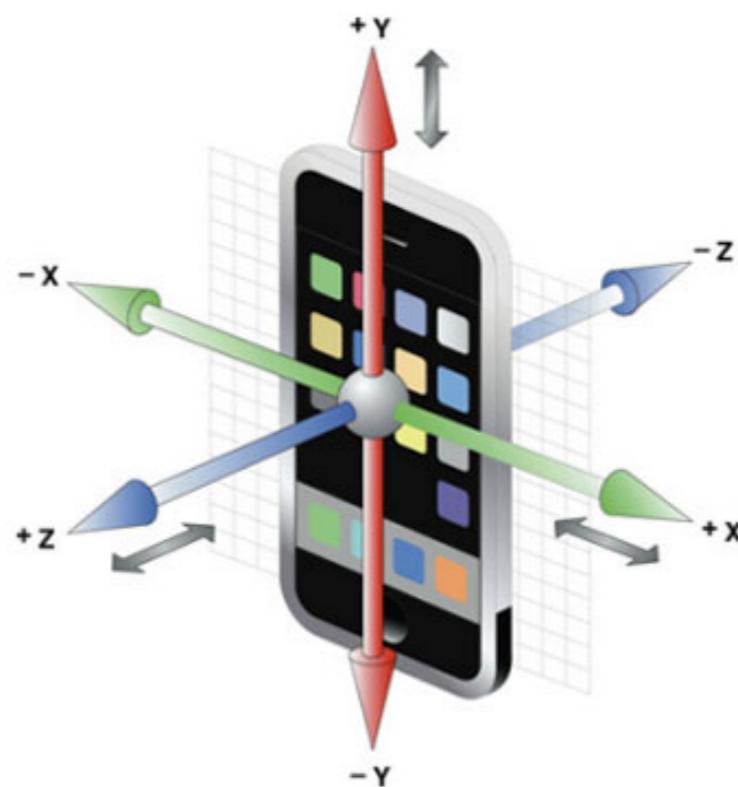
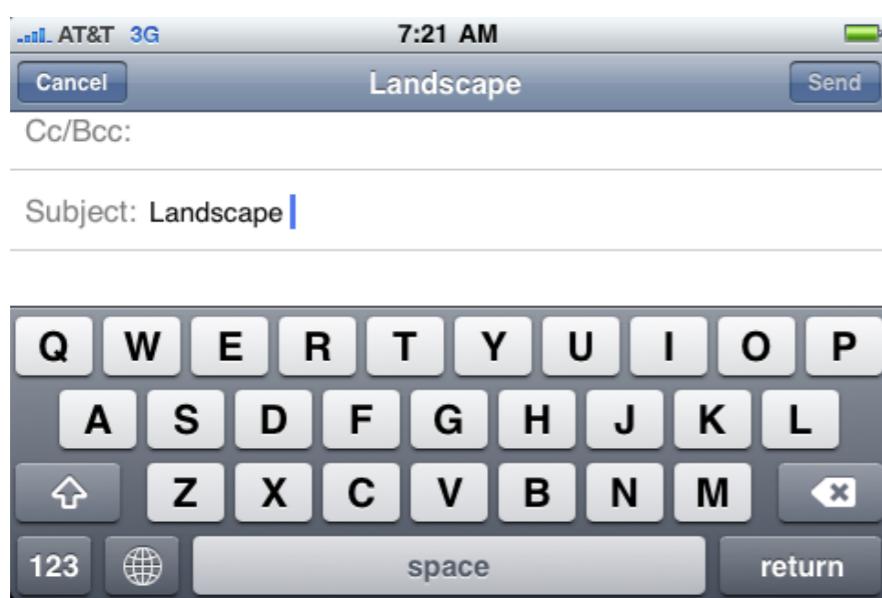
Phone

Mobile Design Principles - Gestures

- One of the most iconic aspects of modern touch interfaces is that they support gesture-based user interaction. As cool as gestures are, there are several things you need to keep in mind:
- **Invisible:** Gestures are invisible, so discovery is an issue. You have to decide how to reveal their existence to the user. A clever approach is on the promotional iPads mounted in Apple's retail stores. When a page first loads, any scrollable areas do a quick 'reverse scroll' into their default position. This immediately invites a swipe or flick gesture from the user without having to explicitly indicate which areas are scrollable.
- **Two hands:** Multi-touch gestures require two-handed operation. This is particularly evident in the native Maps app on iOS which uses a pinch open gesture to zoom out. This can be an annoying limitation. Android addresses this issue by including zoom in/out buttons overlaid on the map
- **Nice to have:** In most cases, gestures are 'nice to have' but not critical. Sort of like keyboard shortcuts – power users will love them, but most people won't even know they are there.
- **No replacement:** A common vocabulary for gestures doesn't exist yet so it's too soon for most apps to skip visible controls that can be used with a single-finger.

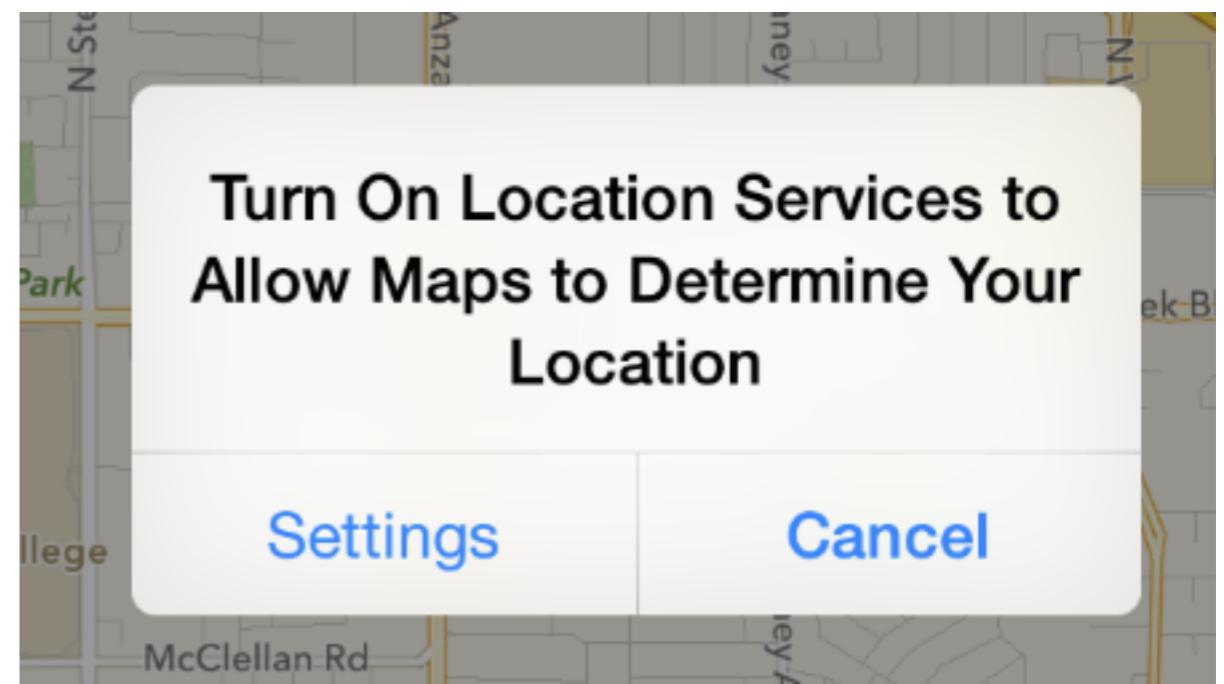
Mobile Design Principles - Orientation

- Portrait is by far the most popular orientation so optimise for this case first.
 - If your app invites lots of typing, you should support landscape orientation so people can access the larger keyboard.
 - When orientation changes unexpectedly, it's, well... disorienting. If you think your app will be used for long periods of time (for example, the Kindle Reader app), consider adding an orientation lock right in the app.



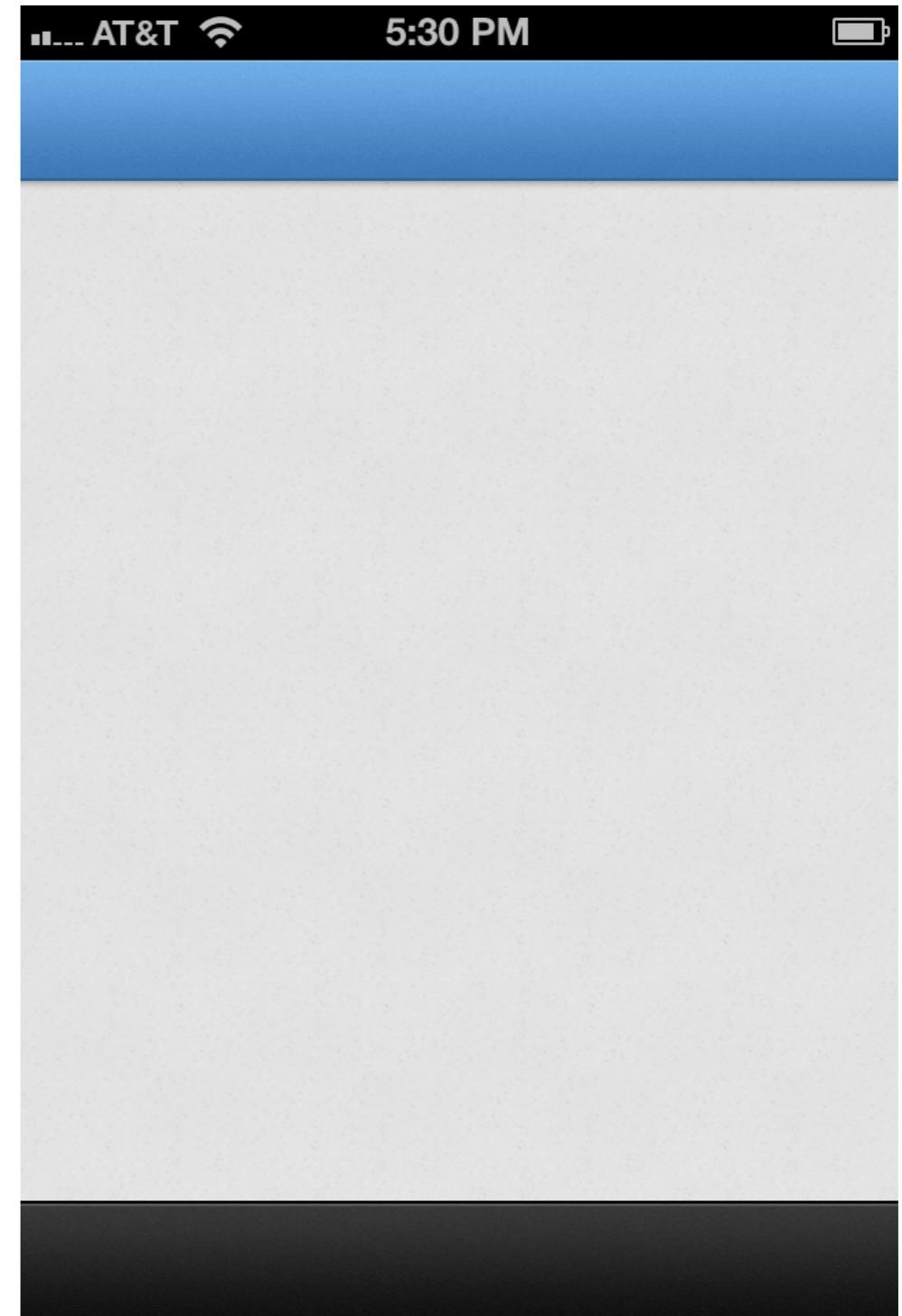
Mobile Design Principles - Communications

- **Provide feedback:** Provide instant feedback for every interaction. If you don't, the user will wonder if the app has frozen up, or if they missed the target they were trying to hit. The feedback could be tactile (like the Android 'thump' vibration), or visual (highlighting a tapped button, for instance). If the user has requested an action that is going to take a long time, display a spinner or progress bar to let them know that you received their request and are working on it.
- **Modal alerts:** Modal alerts are extremely pushy and intrusive to the user's flow, so you should only use them when something is seriously wrong. Even then, try to mitigate the intensity by keeping language reassuring and friendly. Remember not to use modal alerts for 'FYI' type information.
- **Confirmations:** When you have to ask a user to confirm an action, it's acceptable to display a modal confirmation dialog (such as 'Are you sure you want to delete this draft?'). Confirmations are less intrusive than alerts because they are in response to a user action and therefore in context and perhaps even expected. Be sure to make the 'safest' choice the default button in the dialog to help avoid inadvertent destructive actions.



Mobile Design Principles - Launching

- When a user goes back into your app after having used it previously, you should resume operations right where the user left off. This will give the illusion of speed and contribute to an overall feel of responsiveness.
- If possible, the launch screen you display when the app is first loading should be a 'content-less' image of your app. Anything that looks interactive (such as buttons, links, icons, content) will create frustration by inviting failed interactions.
- NOTE: Resist the temptation to place branding materials on your launch screen. They make the user feel as if they're viewing an ad and they'll resent you for making them sit through it. Of course, a branded launch screen doesn't last any longer than an empty one, but the perception of delay exists regardless.



Mobile Design Principles - First Impressions

- **Your icon:** Your icon has to compete for attention in a sea of other icons. That being the case, think of it more as the business card than an art piece. Be literal – show what your app does. Use a strong silhouette and keep text to a minimum. A polished icon suggests a polished app, so it's worth devoting serious time and money to doing it right.
- **First launch:** First launch is a make or break situation. If a new user gets confused or frustrated while trying to acquaint themselves with your app, they'll ditch it ASAP. If your app provides complex functionality, you might want to include a 'tips and tricks' overlay, or perhaps a few panels of orientation screens. Note that this is not a substitute for a good design; if you find yourself creating a lot of help text, it could indicate that your UI needs work.

Human Interface Principles

- As a user, you notice when an app makes it hard to tell whether it received your input, or when a popover seems to emerge from different areas on the screen without apparent reason.
- In cases like these, what you notice is the app's failure to follow the fundamental principles of human interface design.
- In this context, the term human interface refers to the interaction between people and devices, including the software that runs on them. An app (or a device) is easy and enjoyable for people to use when its human interface builds on the ways in which people actually think and behave.

Human Interface Principles

- Apple's human interface design principles codify several high-level aspects of human-device interaction that have a perceptible effect on the user experience.
- As you design your app, keep in mind the following principles of HI design:
 - **Aesthetic Integrity**
 - **Consistency**
 - **Direct Manipulation**
 - **Feedback**
 - **Metaphors**
 - **User Control**

Aesthetic Integrity

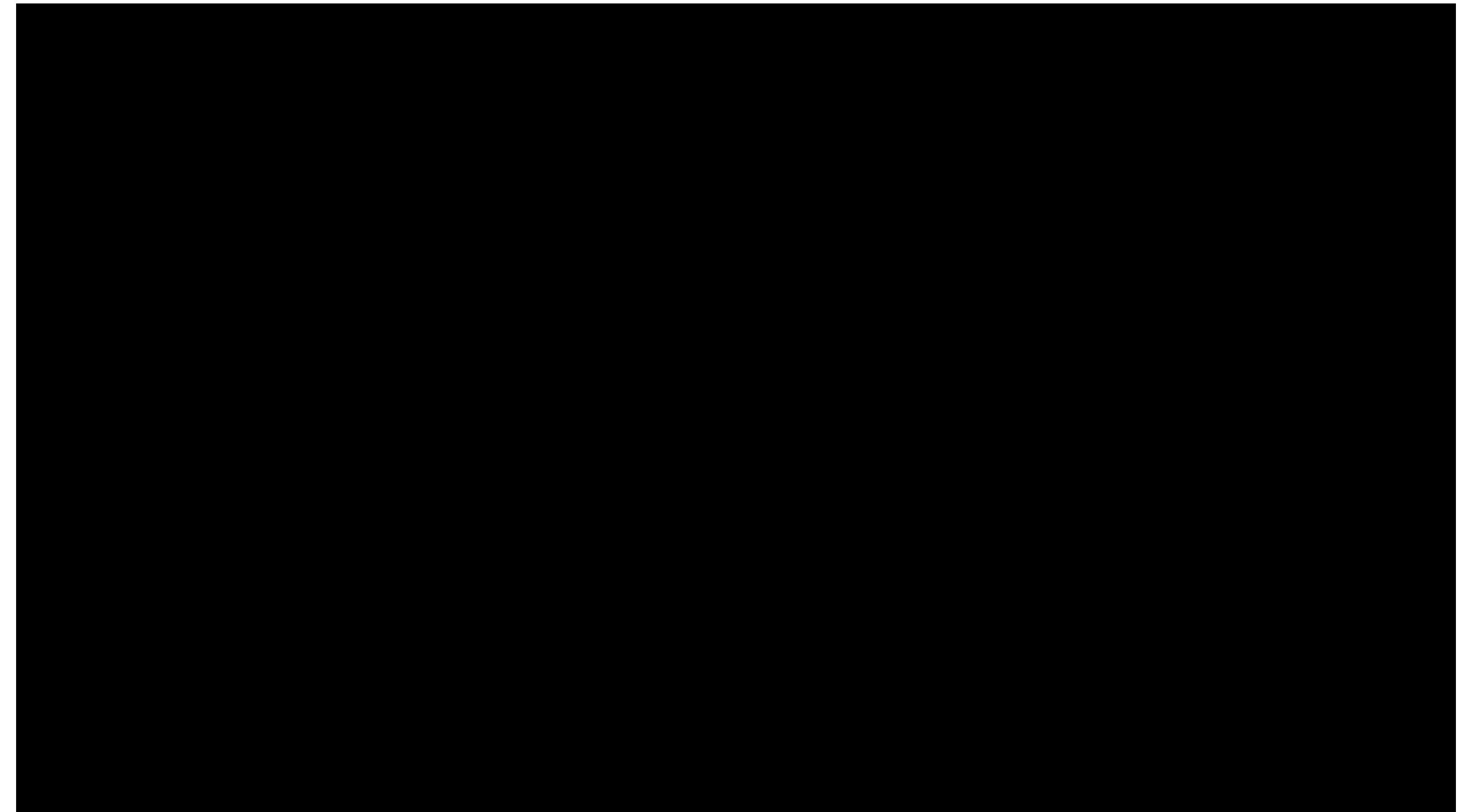
- Aesthetic integrity is not a measure of how beautiful an application is. It's a measure of how well the appearance of the app integrates with its function.
- For example, an app that enables a productive task generally keeps decorative elements subtle and in the background, while giving prominence to the task by providing standard controls and behaviors.
- Such an app gives users a clear, unified message about its purpose and its identity. If, on the other hand, the app enables the productive task within a UI that seems whimsical or frivolous, people might not know how to interpret these contradictory signals.
- Similarly, in an app that encourages an immersive task, such as a game, users expect a beautiful appearance that promises fun and encourages discovery. Although people don't expect to accomplish a serious or productive task in a game, they still expect the game's appearance to integrate with the experience



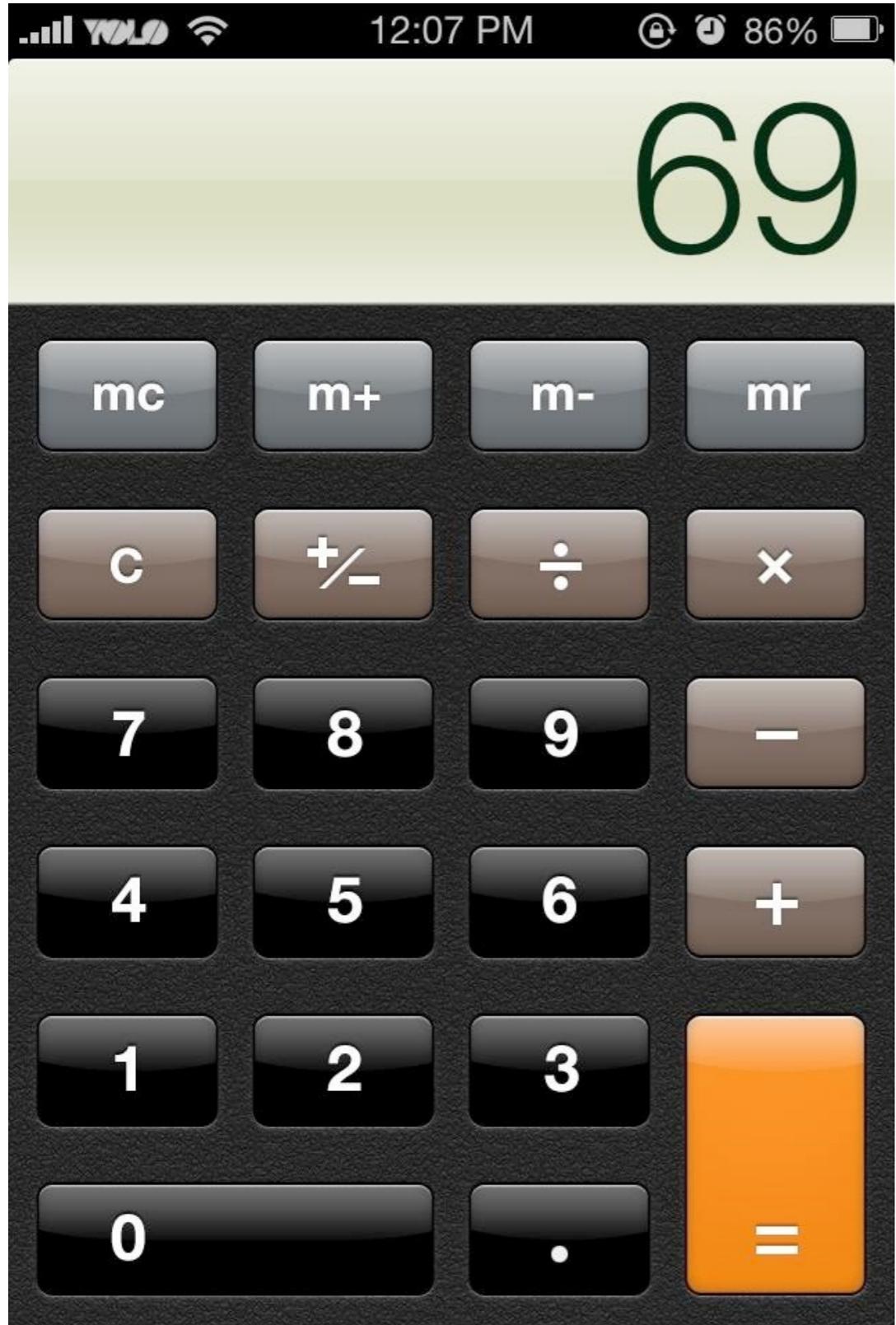
Aesthetic Integrity



iOS6 vs iOS7 - Aesthetic Integrity



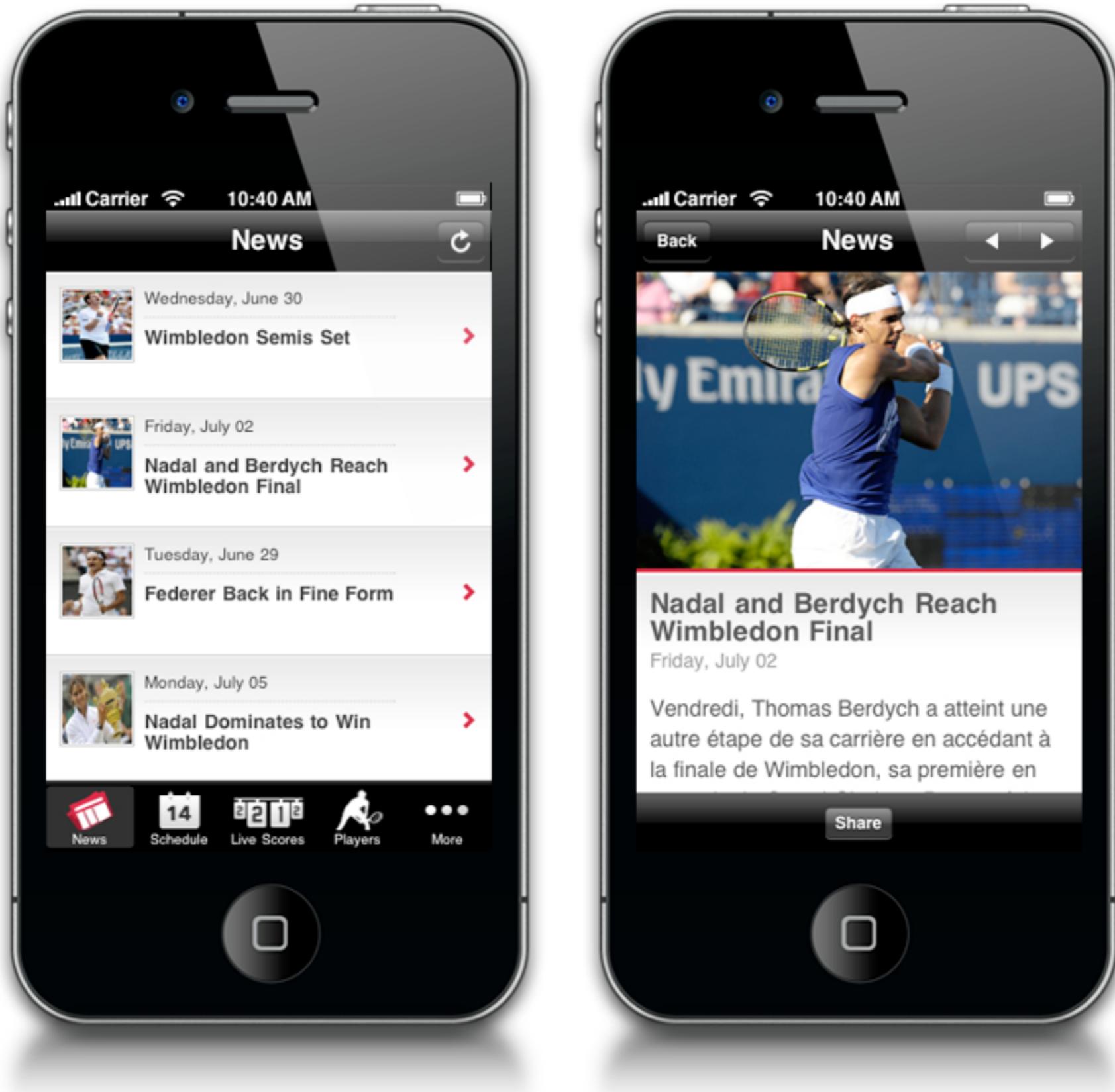
What Is Flat Design? (Tuts+, 2013) [3m 10s]
<http://www.youtube.com/watch?v=skn8pMfbWqY>



Flat Design in iOS 7

Consistency

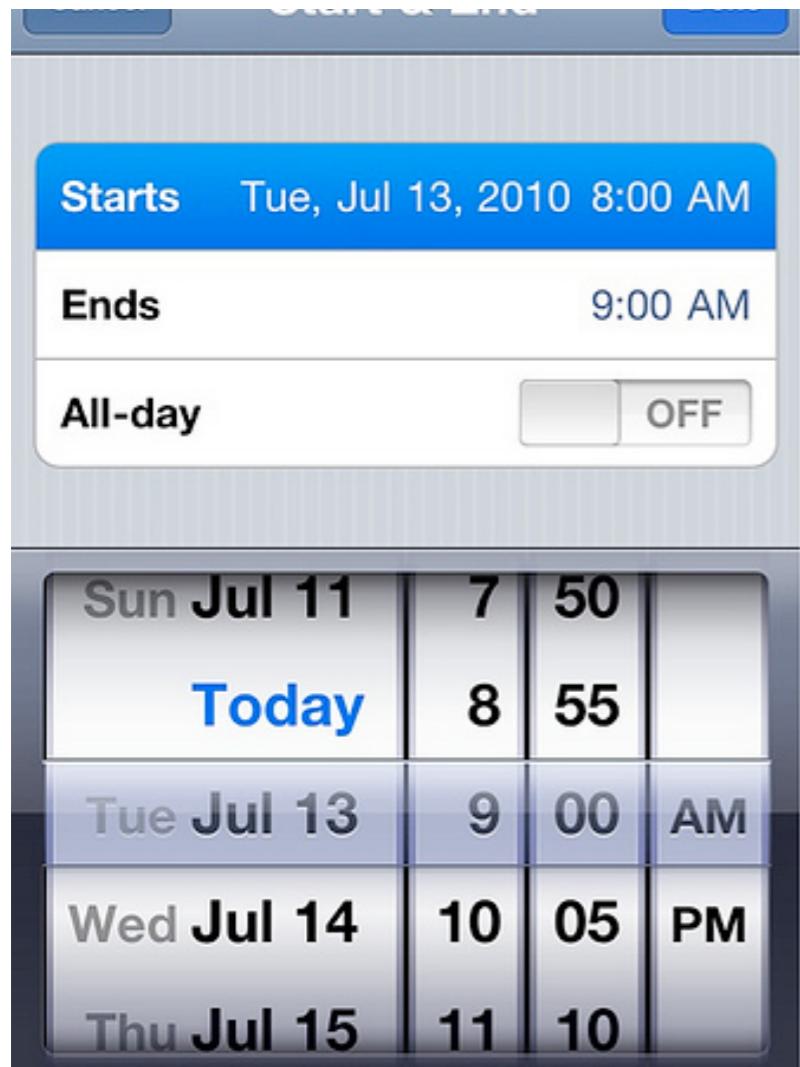
- Consistency in the interface allows people to transfer their knowledge and skills from one application to another.
- A consistent application is not a blatant copy of other applications. But it takes advantage of the standards and paradigms people are comfortable with.
- To determine whether an app follows the principle of consistency, ask yourself these questions:
 - *Is the application consistent with standards? Does it use system-provided controls, views, and icons correctly? Does it incorporate device features in a reliable way?*
 - *Is the application consistent within itself? Does text use uniform terminology and style? Do the same icons always mean the same thing? Can people predict what will happen when they perform the same action in different places? Do custom UI elements look and behave the same throughout the app?*
 - *Within reason, is the application consistent with its earlier versions? Have the terms and meanings remained the same? Are the fundamental concepts essentially unchanged?*



Consistency

Direct Manipulation

- When people directly manipulate onscreen objects instead of using separate controls to manipulate them, they're more engaged with the task and they more readily understand the results of their actions.
- Phone/Tablet users enjoy a heightened sense of direct manipulation because of the Multi-Touch interface. Using gestures gives people a greater affinity for, and sense of control over, the objects they see onscreen, because they're able to touch them without using an intermediary, such as a mouse.
- For example, instead of tapping zoom controls, people can use the pinch gestures to directly expand or contract an area of content. And in a game, players move and interact directly with onscreen objects. For example, a game might display a combination lock that users can spin to open.
- In an app, people can experience direct manipulation when they:
 - Rotate or otherwise move the device to affect onscreen objects
 - Use gestures to manipulate onscreen objects
 - Can see that their actions have immediate, visible results



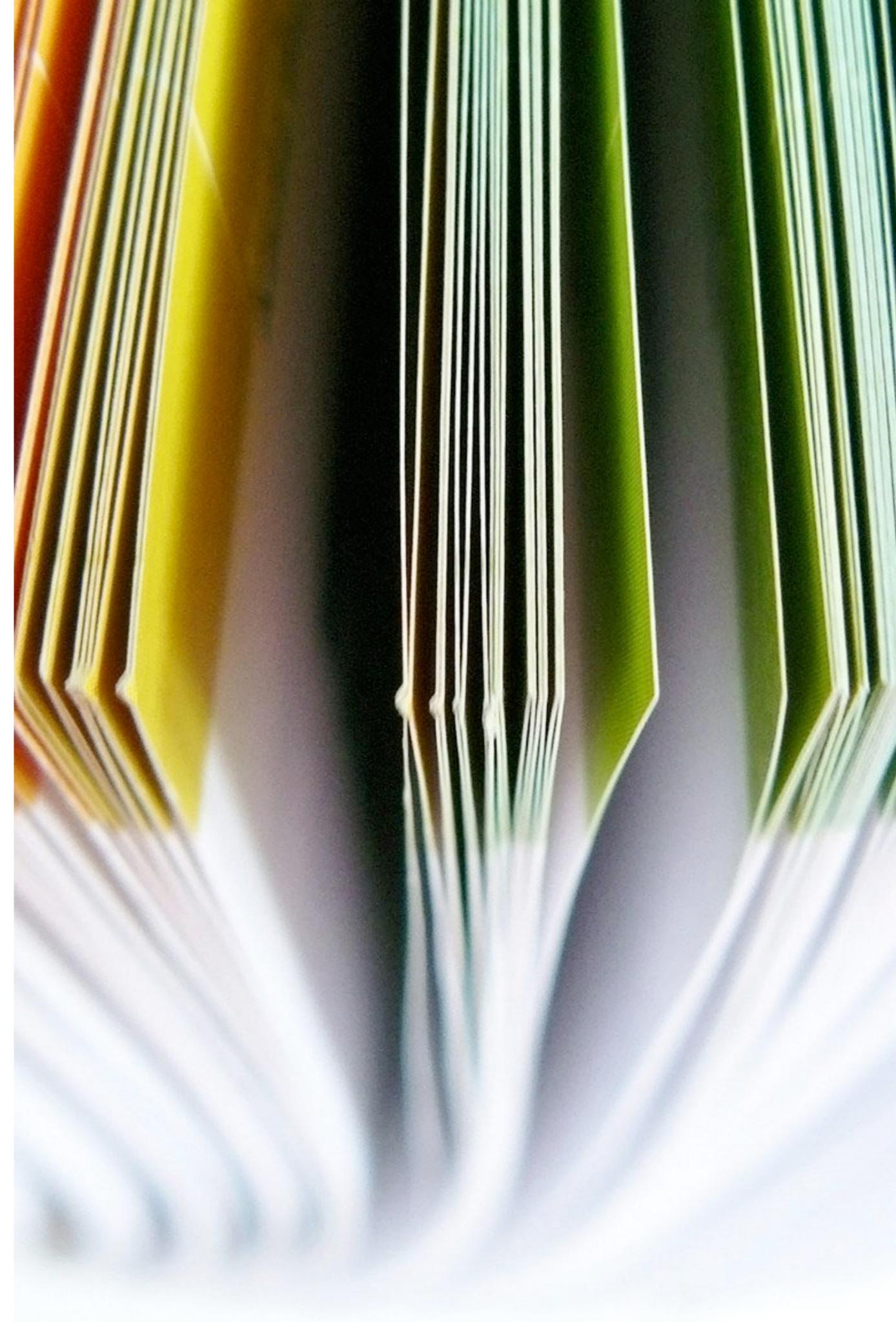
Direct Manipulation

Feedback

- Feedback acknowledges people's actions and assures them that processing is occurring.
- People expect immediate feedback when they operate a control, and they appreciate status updates during lengthy operations.
- The built-in iOS applications respond to every user action with some perceptible change. For example, list items highlight briefly when people tap them. During operations that last more than a few seconds, often an animated progress indicator is shown
- Subtle animation can give people meaningful feedback that helps clarify the results of their actions. For example, lists can animate the addition of a new row to help people track the change visually.
- Sound can also give people useful feedback. But sound shouldn't be the primary or sole feedback mechanism because people may use their devices in places where they can't hear or where they must turn off the sound

Metaphors

- When virtual objects and actions in an application are metaphors for objects and actions in the real world, users quickly grasp how to use the app.
- The classic example of a software metaphor is the folder: People put things in folders in the real world, so they immediately understand the idea of putting files into folders on a computer.
- The most appropriate metaphors suggest a usage or experience without enforcing the limitations of the real-world object or action on which they're based. For example, people can fill software folders with much more content than would fit in a physical folder



Metaphors

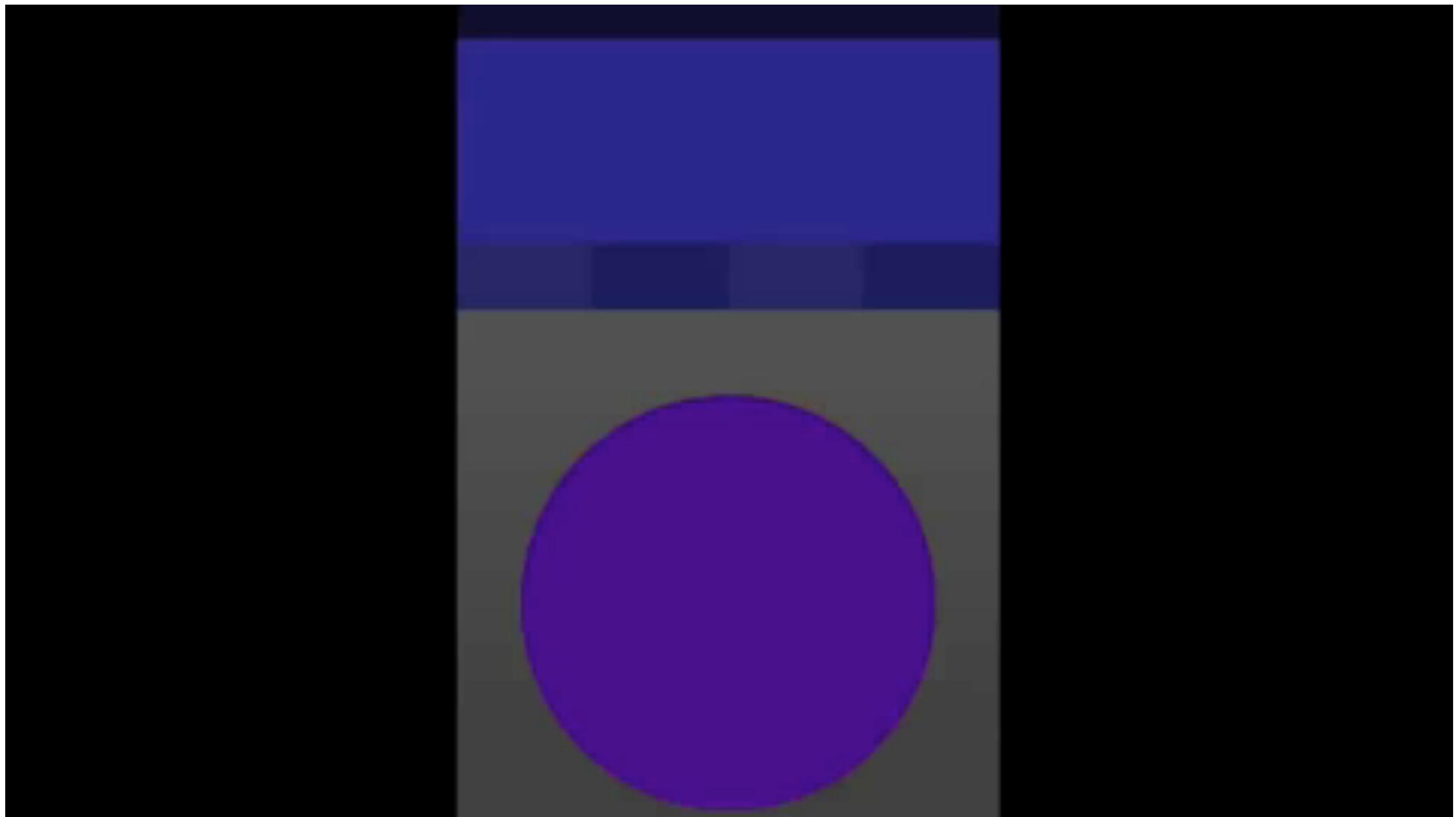
- Apps provide great scope for metaphors because it supports rich graphical images and gestures. People physically interact with realistic onscreen objects, in many cases operating them as if they were real-world objects.
- In general, metaphors work best when they're not stretched too far. For example, the usability of software folders would decrease if they had to be organised into a virtual filing cabinet
- Metaphors in an app include:
 - Tapping music playback controls
 - Dragging, flicking, or swiping objects in a game
 - Sliding On/Off switches
 - Flicking through pages of photos
 - Spinning picker wheels to make choices

User Control

- People, not applications, should initiate and control actions.
Although an application can suggest a course of action or warn about dangerous consequences, it's usually a mistake for the app to take decision-making away from the user. The best apps find the correct balance between giving people the capabilities they need while helping them avoid dangerous outcomes.
- Users feel more in control of an app when behaviors and controls are familiar and predictable. And, when actions are simple and straightforward, users can easily understand and remember them.
- People expect to have ample opportunity to cancel an operation before it begins, and they expect to get a chance to confirm their intention to perform a potentially destructive action. Finally, people expect to be able to gracefully stop an operation that's underway.



User Control



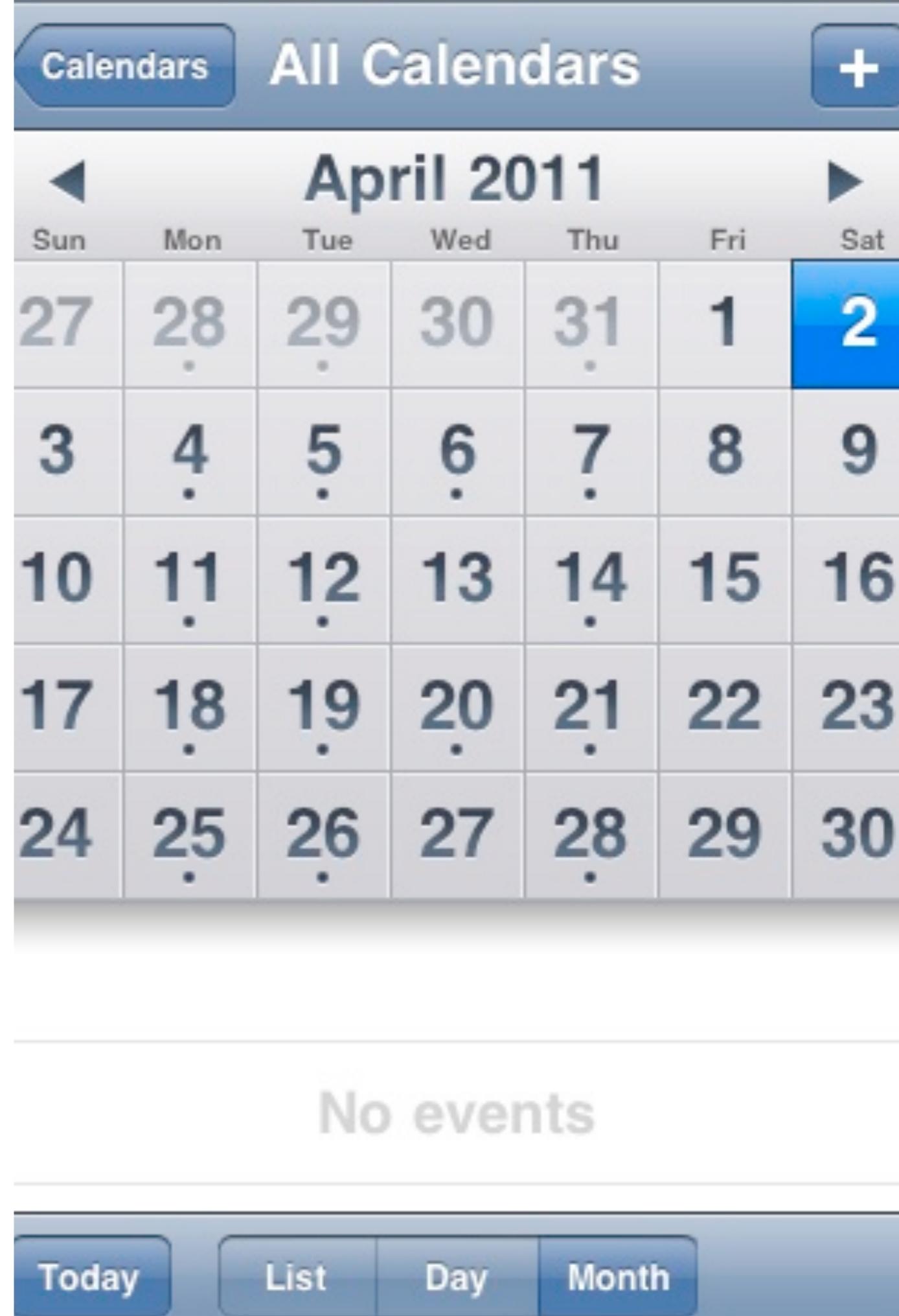
Creating the Interface for Beats (iPhone app) - <http://vimeo.com/4848062>

User Experience Guidelines

- iOS Human Interface Guidelines guidelines that range from user experience recommendations to specific rules that govern the usage of iOS technologies and onscreen elements - **THESE GUIDELINES ARE TRUE FOR ANDROID ALSO**
- Great apps give people streamlined access to the content they care about. To do this, these apps incorporate user experience guidelines such as these:
 - **Focus on the primary task.**
 - **Make usage easy and obvious.**
 - **Use user-centric terminology.**
 - **Make targets fingertip-size**
- **De-emphasize settings.**
- **Use user interface (UI) elements consistently.**
- **Use subtle animation to communicate.**
- **Ask people to save only when necessary**

Focus on the Primary Task

- When an iOS app establishes and maintains focus on its primary task, it is satisfying and enjoyable to use. To maintain that focus, you need to determine what's most important in each context or screen.
- Analyze what's needed in each screen. As you decide what to display in each screen always ask yourself, Is this critical information or functionality users need right now?
- If your answer is no, decide whether the information or functionality might be critical in a different context, or if it's not that important after all



Make Usage Easy and Obvious

- Strive to make your application instantly understandable to people, because you can't assume that they have the time (or can spare the attention) to figure out how it works
- Be consistent with the usage paradigms of the built-in applications. Users understand how to navigate a hierarchy of screens, edit list contents, and switch among application modes using the tab bar. Make it easy for people to use your application by reinforcing their experience.
- Make the main function of your application immediately apparent. You can make it so by:
 1. Minimising the number of controls from which people have to choose
 2. Using standard controls and gestures appropriately and consistently so that they behave the way people expect
 3. Labelling controls clearly so that people understand exactly what they do

Make Usage Easy and Obvious

- In the built-in Stopwatch function (part of the iPhone Clock application) users can see at a glance
 - which button stops and starts the stopwatch
 - which button captures lap times



Use User-Centric Terminology

- In all your text-based communication with users, use terminology you're sure that your users understand. In particular, avoid technical jargon in the user interface. Use what you know about your users to determine whether the words and phrases you plan to use are appropriate.
- E.g. the Wi-Fi Networks Settings screen uses plain language to explain how iOS responds to the user's preference.

< Settings Wi-Fi

Wi-Fi



CHOOSE A NETWORK...

network_name



Wi-Fi_network



Wi-Fi_secure



Other...

Ask to Join Networks



Known networks will be joined automatically.
If no known networks are available, you will
have to manually select a network.

Use UI Elements Consistently

- People expect standard views and controls to look and behave consistently across applications.
 - Follow the recommended usages for standard user interface elements. In this way, users can depend on their prior experience to help them as they learn to use your application. You also make it easy for your app to look up-to-date and work correctly if iOS changes the look or behaviour of these standard views or controls.
 - For an app that enables an immersive task, such as a game, it's reasonable to create completely custom controls. This is because you're creating a unique environment, and discovering how to control that environment is an experience users expect in such applications.
- Avoid radically changing the appearance of a control that performs a standard action. If you use unfamiliar controls to perform standard actions, users will spend time discovering how to use them and will wonder what, if anything, your controls do that the standard ones do not.
 - iOS makes available to you many of the standard buttons and icons used throughout the built-in applications. For example, you can use the same Refresh, Organise, Trash, Reply, and Compose icons that Mail uses on both iPhone and iPad



Use UI Elements Consistently

- When an app correctly uses UI elements, such as buttons and tab bars, users are likely to notice only that the app behaves as they expect.
- But when an app uses UI elements incorrectly, users are often quick to voice their dissatisfaction. Take care to follow UI element usage guidelines. E.g
- Ensure that the back button in a navigation bar displays the title of the previous screen.
 - Always provide feedback when users select an item listed in a table view.
- Don't remove a tab from a tab bar when its function is unavailable.
 - Use system-provided buttons and icons according to their documented meaning.

Proven Design Strategies

- **Distill the feature list**
- As early as possible in the design process, define precisely what your app does and who your target audience is.
- Use this definition (called an application definition statement) to filter out unnecessary features and to guide the style of the app.
- Although it's tempting to think that more features make a better app, the opposite is more often true. The best apps tend to focus on enabling a main task and providing only those features that users need to accomplish the task.
- E.g. Placemark allows users to record info about places you find interesting

Proven Design Strategies

- **Design for the device**
- In addition to integrating the patterns of the iOS user interface and user experience, make sure that your app feels at home on the device.
- If you plan to develop a universal app (that is, an app that runs on both iPhone and iPad), this means that you must design a different UI for each device, even though much of the underlying code can be the same.
- Similarly, if you plan to start with web-based content, it's essential that you redesign the content to look and feel like a native app.

Proven Design Strategies

- **Customise appropriately**
- Every app includes some UI customisation, if only in its App Store icon.
- The iOS SDK gives you the ability to customise every aspect of the UI, so it's up to you to decide how much customisation is appropriate. The best apps balance customisation with clarity of purpose and ease of use.
- Ideally, you want users to recognise the distinctiveness of your app, while at the same time appreciating the consistency that makes an app intuitive and easy to use.

Proven Design Strategies

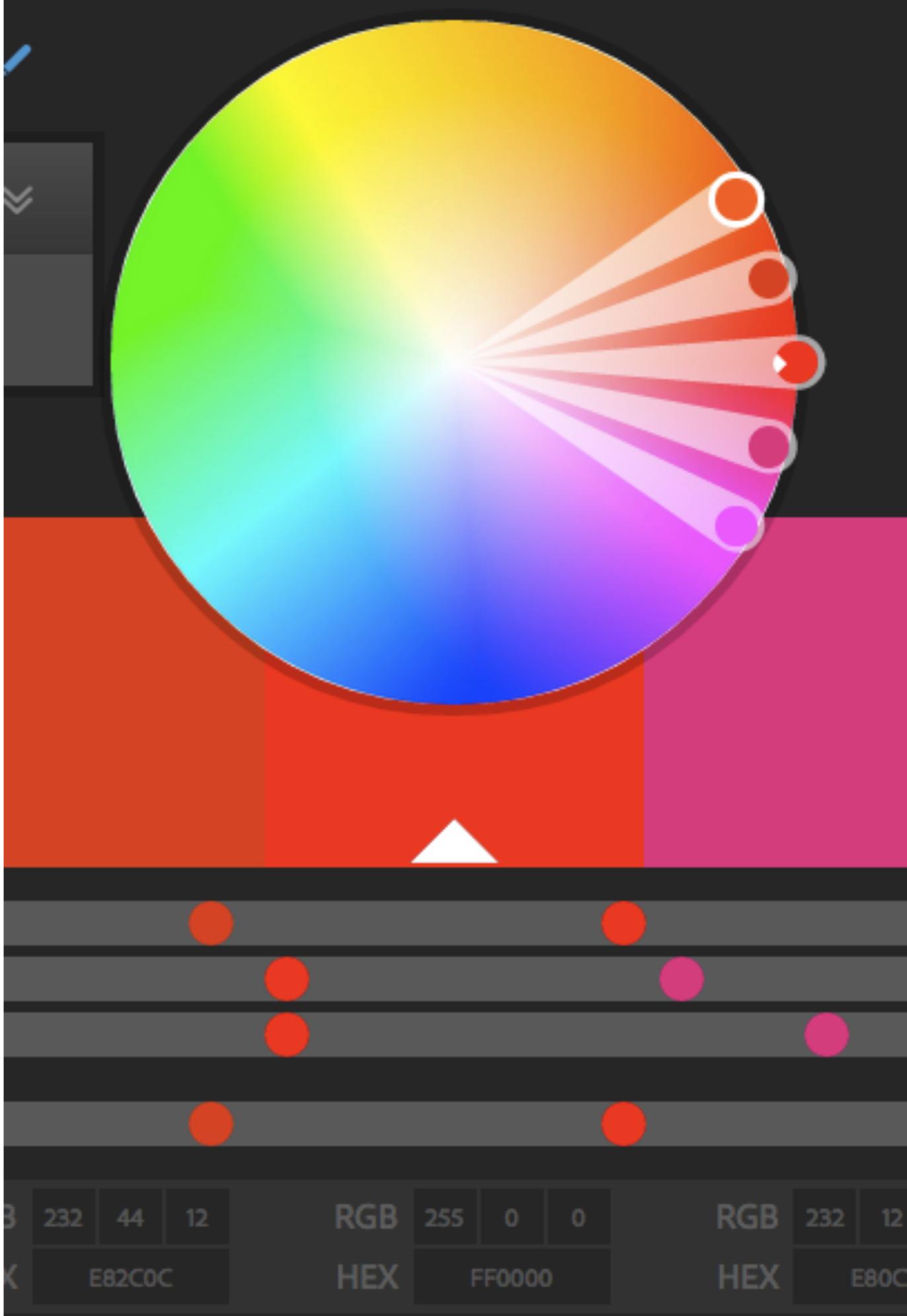
- **Prototype and iterate**
- Soon after you've decided what features to include, begin creating testable prototypes.
- Early prototypes don't need to display real UI or art and they don't need to handle real content, but they do need to give testers an accurate idea of how the app works.
- During testing, pay particular attention to what testers try and fail to do, because those attempts can reveal places where the app appears to promise a behaviour that it doesn't deliver. Continue testing until you're satisfied that users can easily grasp how the app works and operate all its features.

Practicalities

- Android devices come in all shapes and sizes, so your app's layout needs to be flexible. That is, instead of defining your layout with rigid dimensions that assume a certain screen size and aspect ratio, your layout should gracefully respond to different screen sizes and orientations.
 - Use view dimensions that allow the layout to resize
 - Create alternative UI layouts according to the screen configuration
 - Provide bitmaps that can stretch with the views
- Detailed information on these issues (and more!) can be found at <https://developer.android.com/training/multiscreen/screensizes>

Practicalities

- Sympathetic Colour Schemes - <http://color.adobe.com/>
- Icon Shock Free Collection - www.iconshock.com/icon_sets/
- Stock Photos - www.stockfreeimages.com/
- UIKit User Interface Catalog - <http://bit.ly/1mPNkK6>
- Balsamiq wireframe mockups - <http://balsamiq.com/>
- iOS Tools & Resources for Designers - <http://bit.ly/1eUi5DM>





Sketching iPhone User Interfaces - <http://www.youtube.com/watch?v=mzj902oaCRQ>

Summary

- Design With The User In Mind
- Human Interface Principles
- Mobile Design Principles
- Practicalities



More Information

- Apple's iOS Human Interface Guidelines https://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html#/apple_ref/doc/uid/TP40006556-CH1-SW1
- Apple Developer Videos - Designing User Interfaces for iPhone and iPad <https://developer.apple.com/videos/ios/>
- Getting Started in iOS User Interface Design <http://webdesignledger.com/tips/getting-started-in-ios-user-interface-design> - ** has links to lots of free image resources **

Bibliography

- Alan Cannistraro, ‘iPhone Application Development’, Stanford University, iTunes U, 2010
- Apple Developer Programme. <http://developer.apple.com>
- Aaron Hillegass, ‘Cocoa Programming for Mac OS X’ (3rd Edition), Addison Wellsey, 2008
- Stark, Jonathan, “The 10 principles of mobile interface design”, Creative Bloq. 15 April 2012. <http://www.creativebloq.com/mobile/10-principles-mobile-interface-design-4122910> Access 22 January 2015
- Joe Conway & Aaron Hillegass, ‘iOS Programming’ (3rd Edition), Big Nerd Ranch. 2012
- Ray Wenderlich, Tutorials for iOS Developers. www.raywenderlich.com
- Tuts+, “Mastering Flat Design: What is Flat Design?” 12 Aug 2013. <https://www.youtube.com/watch?v=skn8pMfbWqY> Access 21 January 2015