THE APP DESIGN HANDBOOK

BY NATHAN BARRY

Learn to design beautiful iOS Applications.

TABLE OF CONTENTS

Please do not distribute or share without permission. If you have questions get in touch. My email address is nathan@thinklegend.com. I try to respond to all the email I receive.

Section 1

THE IMPLEMENTATION

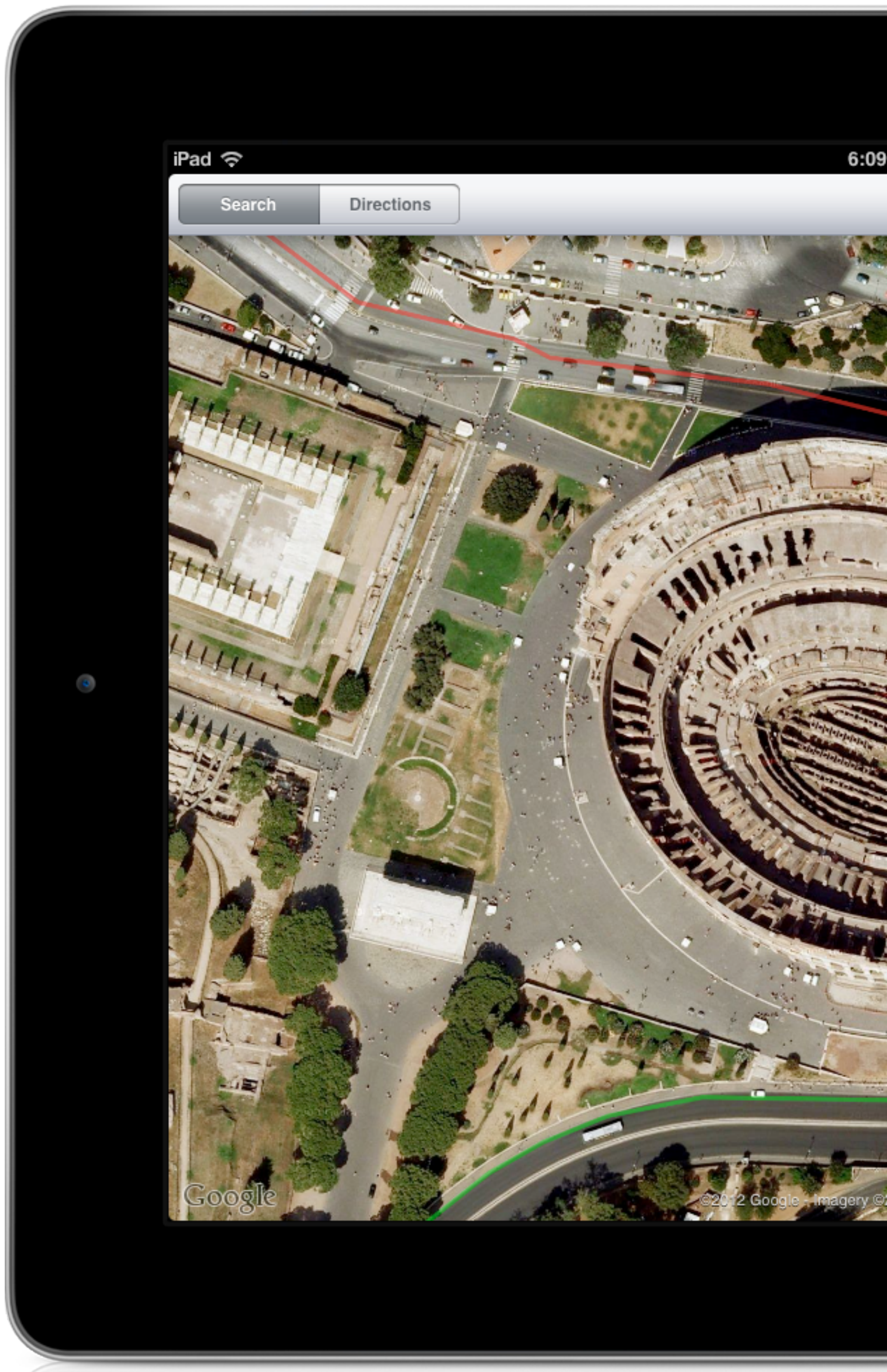
"When technology gets out of the way, everything becomes more delightful--even magical."

- iPad 2 commercial

As corny as it may sound at first, that is what you are trying to create: a magical experience. When I first heard Apple using "magical" to describe the iPad, I laughed. Apple has always been famous for using powerful, even exaggerative, vocabulary in their marketing, but this time I really felt like they had stopped being serious. That is until I saw how the iPad is used and what developers have created.

Watch a parent’s face as they see their child with autism "speak" for the first time using an app like OneVoice on the iPad. That's magical.

Play with the built-in maps application. Set it to the satellite view and start exploring. The weekend the iPad was released my wife and I spent a long time laying on the couch looking through places we'd already visited, and exploring far-away cities we planned to visit on future trips. It wasn't the technologies or features that made this experience happen. It was already possible with Google Maps on the computer. But the interaction would be completely different. The portable size and touch screen allows for a personal experience in a way that a computer can't. Pinch, pan, and swipe gestures allow you to interact with the maps in a new way. The technology isn't magical, but the experience it



creates is. "When technology gets out of the way," that's where the iOS devices shine.

> "What your users really care about is what your app enables them to do and how it makes them feel when they're using it."
>
> *- Mark Kwano, User Experience Evangelist at Apple*

Application developers hear a quote like this and focus on the first part: "What your users really care about is what your app enables them to do...."

That's easy. We can write down a feature list and start programming against it. Check each feature off as you include it in the application, then submit your app when the list is complete. Yes, you created a useful application, but you didn't create an experience.

You see, the second part of that quote is just as important as the first: "... and how it makes them feel when they're using it." That's the experience. Imagine a maps application for the iPad that required tapping on-screen arrows in order to slowly navigate around the screen, then using a magnifying glass button to zoom in and out. The app is still functional but the joy in using it is completely gone. You've entirely changed the experience, and in doing so changed how your application makes the user feel.

Now that you have that idea in place it is time to focus on the experience. Having a unique idea is less important than creating a magical experience for your users. Let's discuss a few things to improve the experience of nearly any application.

Use intuitive gestures rather than buttons

Buttons have their place, and replacing all of them is not a good idea. But our maps example is a perfect case for gestures being far better than their button counterparts on desktop software. Think carefully about replacing a button with a gesture. If it is intuitive (this is easy to test, we'll cover that later), then use the gesture.

Use animation to show feedback

Real-world objects never jump instantly between two places. In most cases software shouldn't either. In the Mail application an email message will animate from one folder to another if you tell it to move. This provides visual feedback that the email went into the correct folder. It also adds subtle details to the experience that help to make it great overall.

Mimic real-world objects (when appropriate)

The brilliant display on the iPad is just begging to be turned into anything you want it to be. By using textures and representations of objects from the real world you can create on screen an experience that users are already familiar with.

We'll cover the details of each of these in later chapters. For now the most important thing you should learn is that for each iOS application you should be crafting an experience for your users. Maybe it will even feel magical.

iPAD, iPHONE, OR BOTH?

Usually when asked what devices they want to support, the idea person will say “all of them!” But each new device you add takes more time, energy, and money to support. So start with the device that makes more sense and consider expanding from there.

First it is important to know that all iPhone apps will run on the iPad without any additional effort. Unfortunately, it will be at a small size, surrounded by a large black area on the iPad display. There is a button in the lower right to double the size (2x), but that just pixelates the interface.

So there are four options for your app:

- Designed for iPhone (displays poorly, but functional on the iPad).
- iPad only (will not run on iPhone).
- Universal (a single app binary that includes interfaces for both the iPad and iPhone. Purchase it once, use it everywhere).
- Two separate applications (your users will need to purchase separate copies if they want it on their other device as well).

Paper by FiftyThree is a drawing and painting application that takes advantage of the large display on the iPad. Because of this they only released an iPad version, ignoring the smaller-sized devices. Is a drawing app going to be that great on an iPhone screen? Probably not. Focus on the device where you can create the best experience.

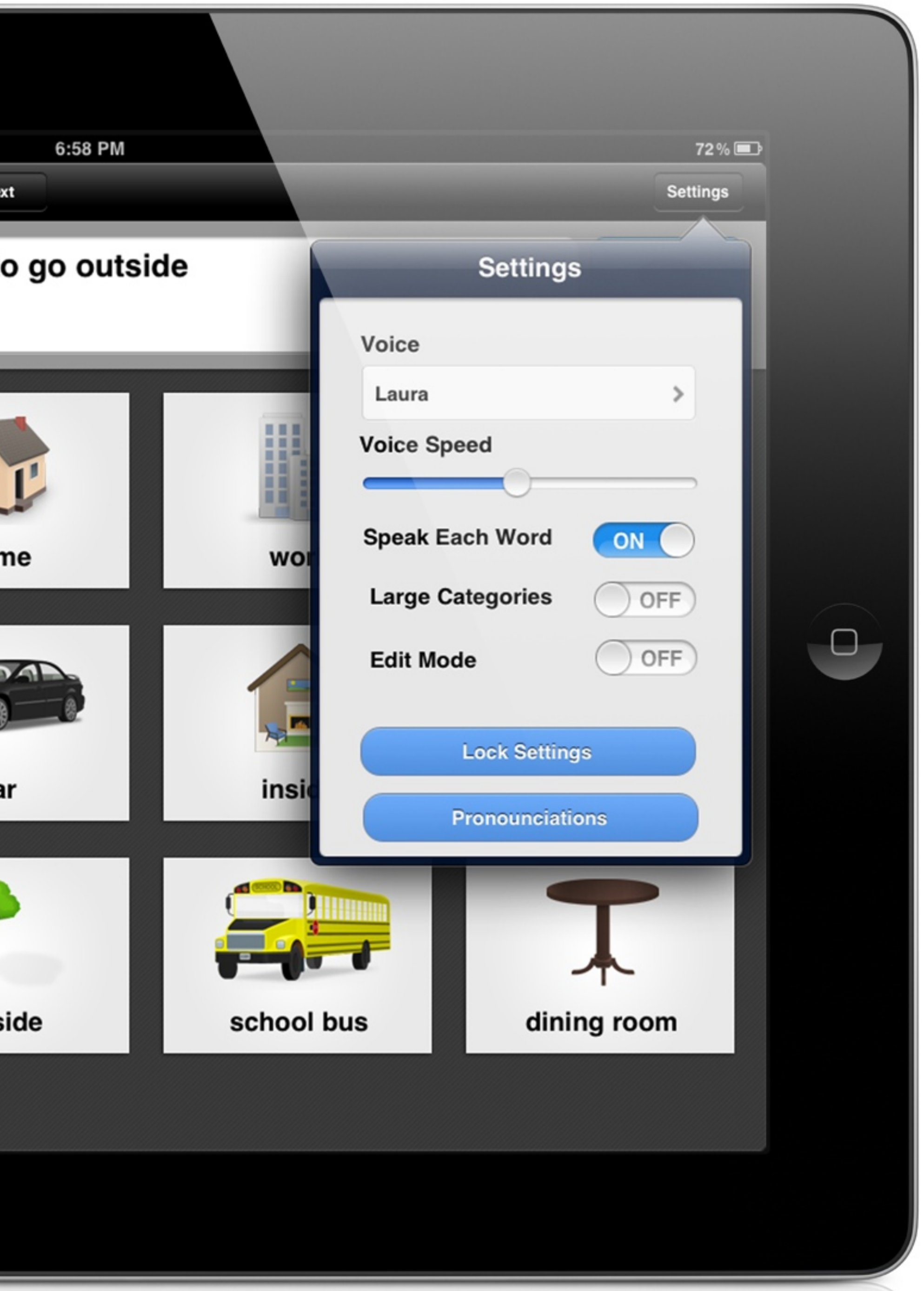
By contrast, Commit, an app to help you form and keep habits, has very simple functionality, but also needs to be kept close to the user so that the reminder messages are received. Though it could be developed as a universal app (designed both for the iPhone and iPad), an iPad-specific interface doesn't add much value.

Instead we can just let the iPad users run the iPhone app at 2x resolution. It isn't the best experience, but it works. And in this case I don't think it is worth the effort to design another version for the iPad.

Judge your app and decide if the added work improves the experience enough to be justified.

Section 3

PREFERENCES & SETTINGS



Settings

Voice: Laura

Voice Speed

Speak Each Word: ON

Large Categories: OFF

Edit Mode: OFF

As you design an application you will inevitably make design decisions that everyone on your team may not agree with. In trying to make our app as great an experience as possible there will be disagreements on the best implementation. Or sometimes the majority of your users will use the app a certain way, but with a few tweaks it could serve a new audience. This is where preferences and settings come in.

In OneVoice, my speaking application, every time you tap on a phrase it is added to the sentence box. Then when you tap Speak, that phrase is spoken. This works great for most users, but after observing children use the application I realized that another option would be needed. Most of the time the kids wouldn’t build a sentence, but instead just tap a single phrase. If you asked them what they wanted to eat they wouldn’t take the time to tap out “I want” + “to eat” + “an apple”, but instead just find “an apple” and tap only that. So I added a setting called Speak Each Word. When this setting is turned on each phrase will be spoken as it is tapped, rather than having to tap the speak button every time.

With this setting OneVoice works for advanced users who want to form a complete sentence before speaking, as well as beginners who just need to select a single word or phrase to convey meaning.

Through settings you can design an application that meets a broader range of uses and fits the needs of more people.

Too Many Settings

You know those apps you use that have all kinds of buttons, cluttered screens, and are so complicated you can barely use them? Often the culprit is too many settings. Those options that the designer added to help make the app usable and customizable for more people are actually what cripples it. With this in mind we need to find a balance.

Cabal Sasser, co-founder and designer at the Mac software company Panic, says that every time you add a preference you are forcing a decision on the user. In many cases adding a setting is reflective of designer laziness. Rather than research and design the best solution you just add a preference and force the decision onto the user. Should the navigation be on the left or right? I don't know. Make it a setting and let the user decide.

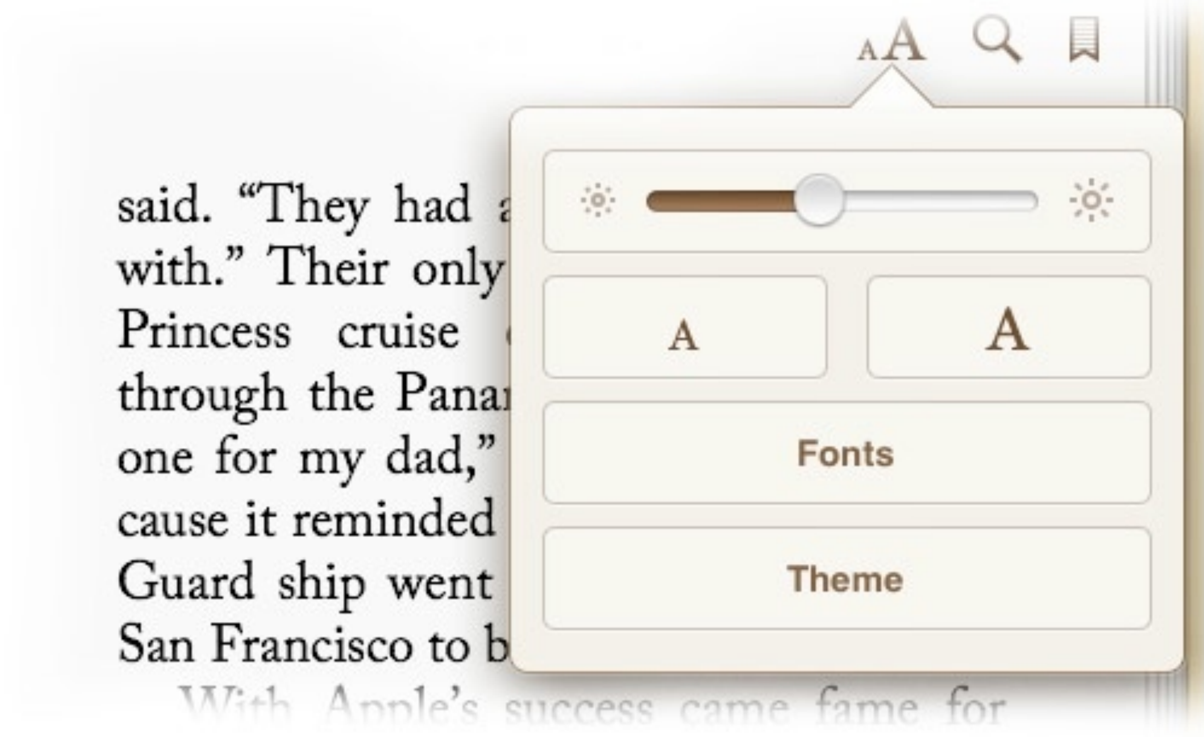
Obviously, I don't think every setting is bad. I add them to my applications. But you do really have to think carefully about adding just the right settings and nothing more.

Where should the settings go?

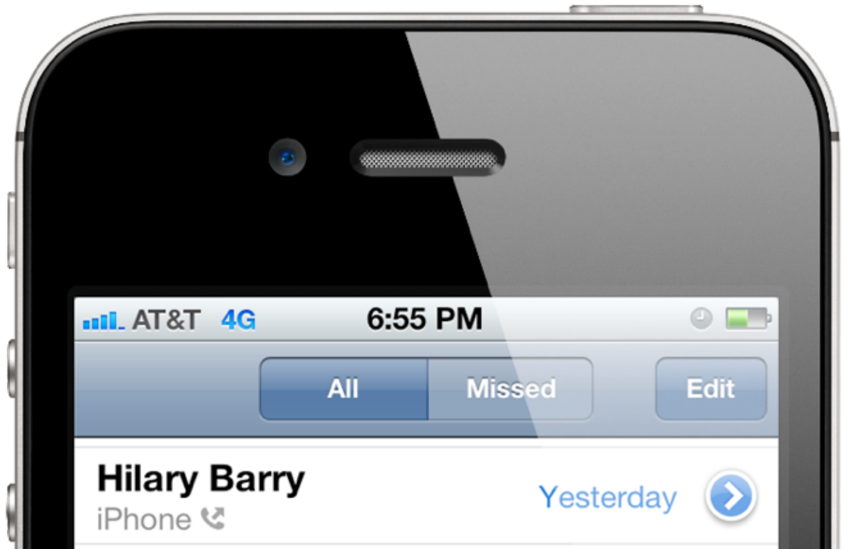
When placing settings within your app you have three basic choices: inline, on a settings page in-app, or separately in the Settings app.

Inline Settings

In iBooks text size needs to be quickly customizable based on the user's preference. Because of this Apple made the text-size setting readily available while reading. This is an inline setting. By placing the setting right next to the objects it controls, the context is very clear and changes can be made very quickly.

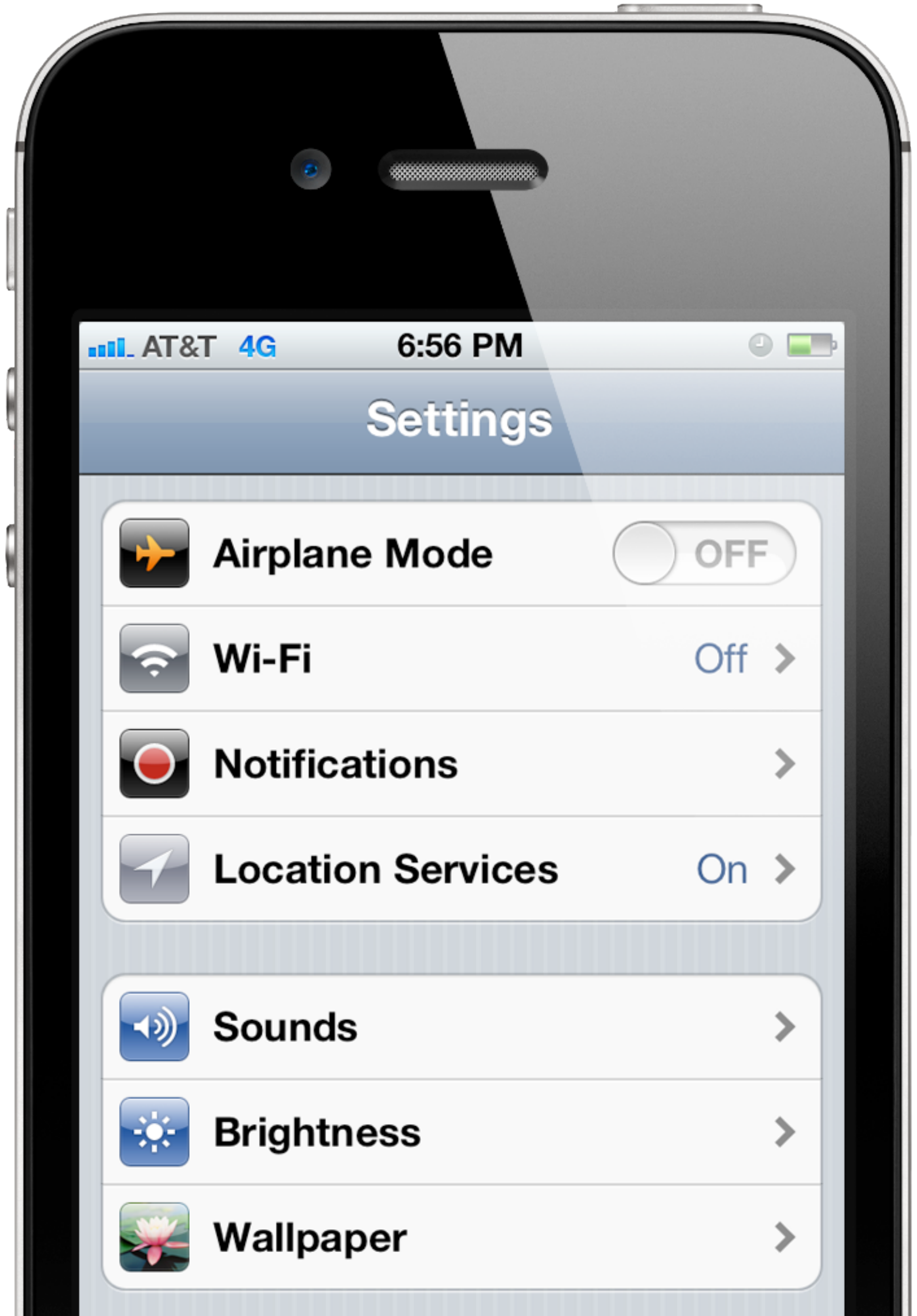


When viewing the recent phone calls on your iPhone there is a setting at the top of the screen. Do you want to see all calls or just missed calls? With the segmented control in the navigation bar you can quickly filter the list to your preference.



In-app Settings Page

If you have several settings that affect the general functions of the app, but don't closely relate to a specific screen, it is common to have a dedicated settings screen inside the app. This could include account functionality, interface preferences, a log-out button, and anything else your app needs. By separating this functionality into its own screen we are able to make sure the rest of the interface does not become cluttered while still allowing the settings to be accessed relatively quickly.



Airplane Mode OFF

Wi-Fi Off

Notifications

Location Services On

Sounds

Brightness

Wallpaper

The Settings App

iOS has a dedicated settings app that includes everything from wi-fi networks to screen brightness. As an app developer you can add settings screens for your app inside of the Settings app. This is great for settings that aren't commonly used that you want to keep hidden from the majority of your users. Because they are in an entirely separate application they aren't likely to be discovered easily. If you want to accommodate a small minority of users with a preference without cluttering your interface, this is a great way to go. Those users can be told about the preferences inside the Settings app, and the users who don't need them will never know about it.

Also the API to add a screen is quite simple. You won't be able to customize the look or control styles as much as you could if they were within your app, but it does have a consistent look across the Settings app.

Here are a few questions to ask yourself while deciding where to place your settings:

- How often will this setting be changed?
- Does it directly affect an element on a specific screen?
- What portion of your users will use this setting?

Text size in iBooks will need to be changed often if different users are sharing the same device, as it directly affects the reading screens. Users will want to see their changes immediately to tell when they have selected an appropriate size. And finally, quite a few users will use this setting, so it should be an inline setting.

See how our test works? Try it with each of the preferences you plan to include in your application. But first you need to decide if the setting should even be included in the app. Too many settings will cripple the app experience, so set the bar very high.

Reasonable defaults

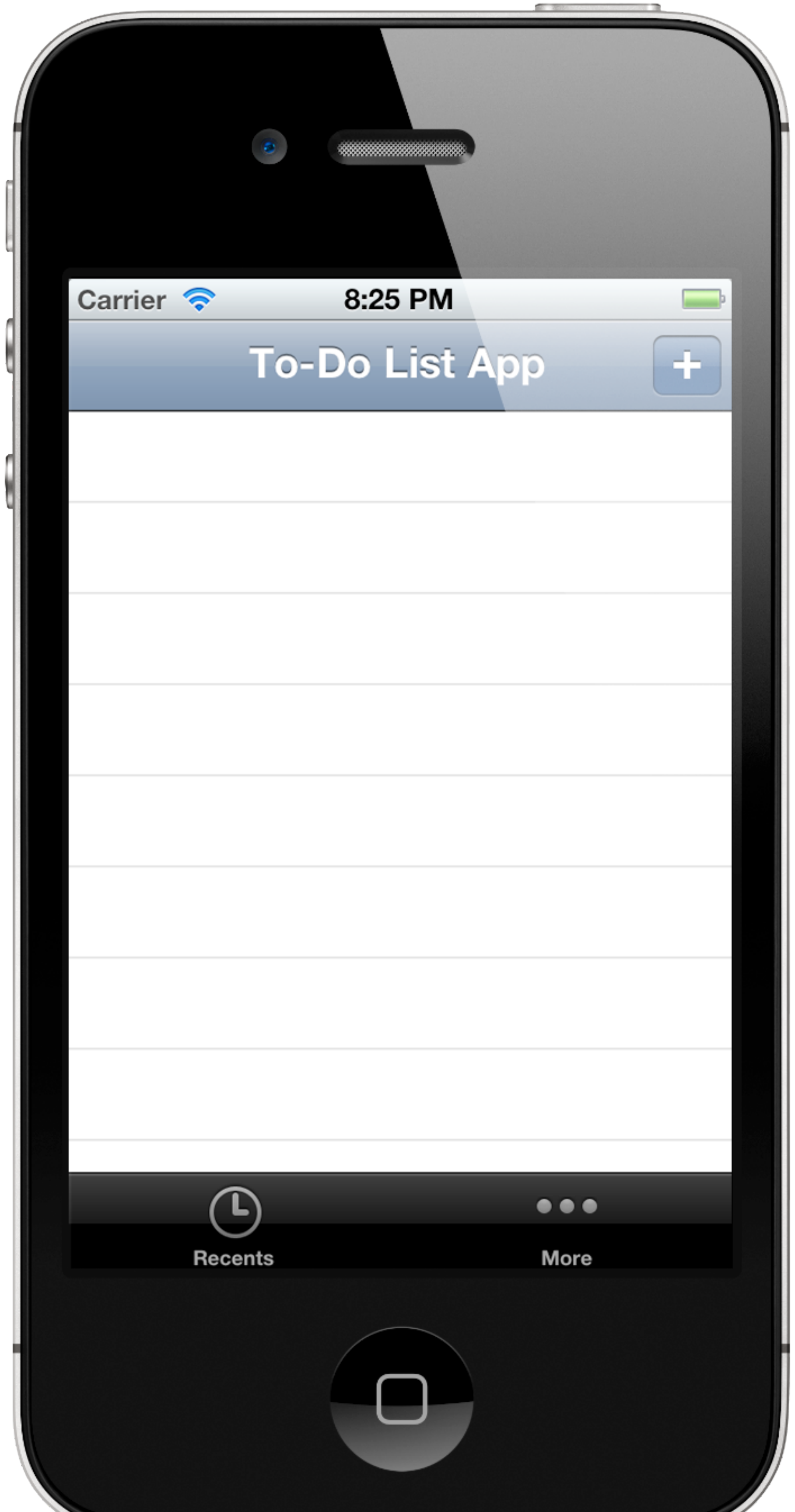
Imagine launching an app for the first time and it takes you through a series of steps to configure the app and set it up just right for you, just like an installation wizard on older versions of Windows. This is a terrible experience. As a designer your goal should be to get the user actually using your application as quickly as possible, not doing a bunch of configurations.

The first problem is that we include too many settings in our apps. Then the second problem is requiring the user to fill out those settings up front. Instead you should provide reasonable defaults for everything possible. Then the user only has to make a change if the selected default doesn't work for them, allowing the user to make fewer decisions before your app is useful to them.

In OneVoice the users can choose between different voices to change the style of speech. Each voice has its own style. Rather than require the user to select a voice before using the app, I chose one for them. Heather, the name of the default voice, is an adult female. If the user of the app is male they will want to select a new voice that better matches their speaking style, but I don't force a choice before the app is usable.

Think of a reasonable default for each of your settings and don't require your users to make a choice. Then your app will work out of the box and the users can decide if they want to make customizations.

DESIGNING THE BLANK SLATE



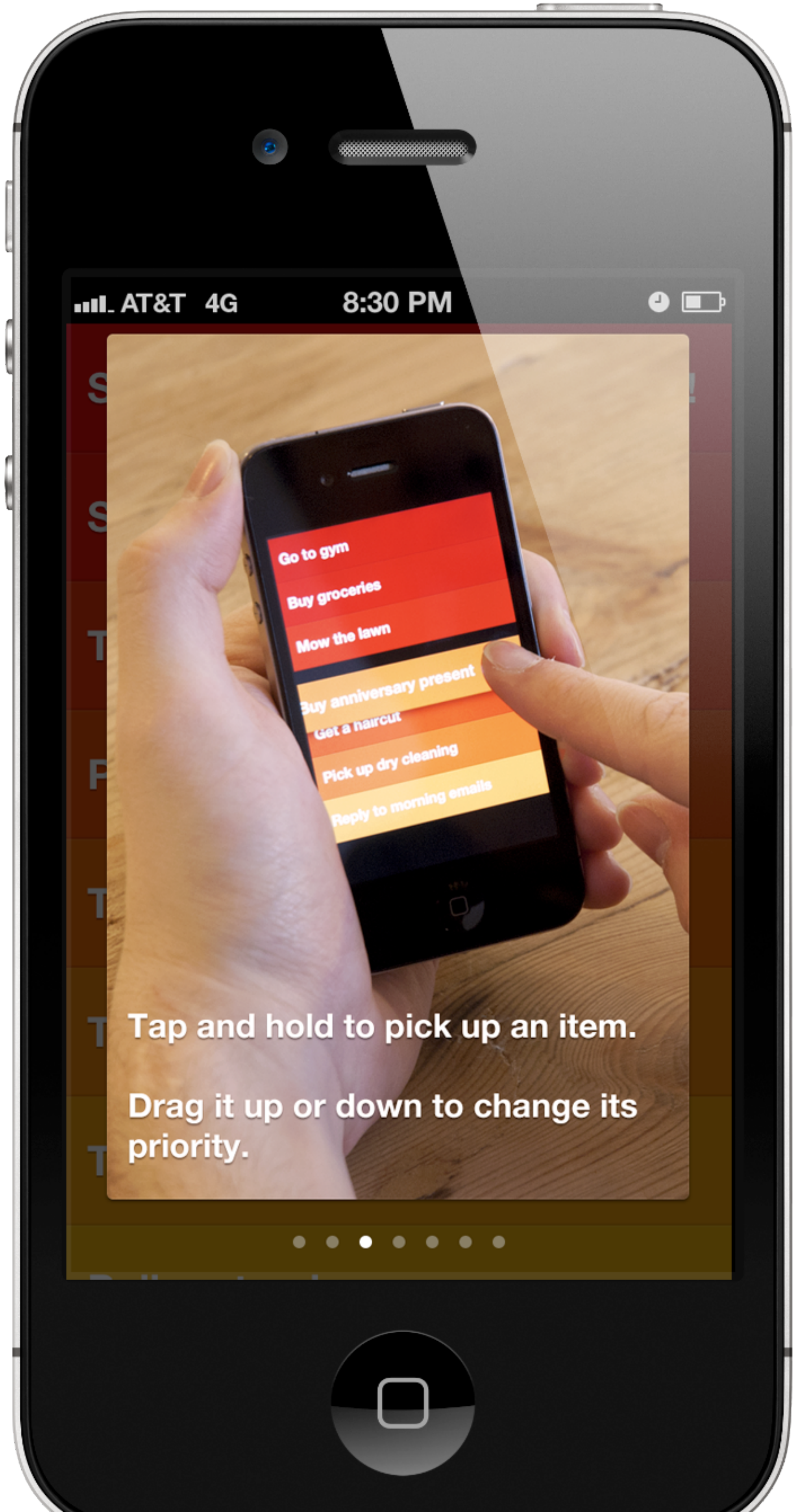
The application loads for the first time. What do you see?

Some applications have real content to show right at the first launch. But many, especially productivity applications, don't have any content until it is added by the user. It is far too common to launch a new app for the first time and encounter a blank screen with an empty view leaving you clueless as to the next step.

Really? That's the best you can do?

This is called the **blank slate**, any screen that is displayed to the user before it has content. How you design this interaction sets the tone for the relationship between the user and your app. If you can't show them content right away, start by teaching.

Use an arrow or brief message to show them the button to add a new item. From there walk them through the process of getting started, explaining the application features and principles along the way. Even just a simple message saying, "Your to-do list is empty. Tap the + icon above to add a new item to your list" is simple and easy, but vastly improves that first-run experience.



Tap and hold to pick up an item.

Drag it up or down to change its priority.

If your application has complex interactions you may need to teach the user about them. Clear is a to-do list app that doesn't have any buttons (other than the keyboard). Instead the app requires you to learn new, but fun, gestures. They teach you how to use the app right away with a series of help screens. Without this information it would take a lot of guessing before the user figured out the gestures. But once they are shown, navigating the app becomes easy and natural.

Whenever your app would display an empty screen think about what you could teach the user in that moment. Take advantage of the opportunity.