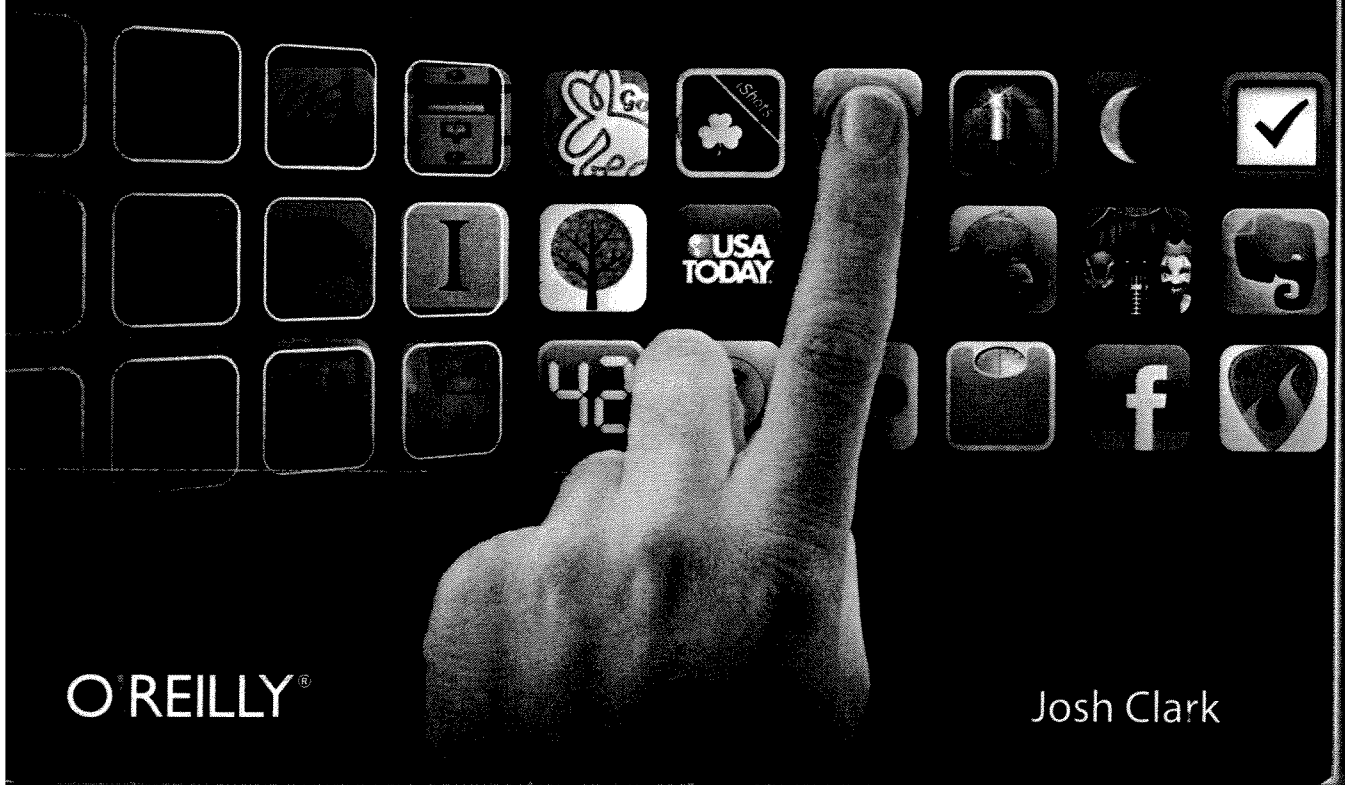
"I'm blown away by Josh Clark's deep understanding of the iPhone user experience. This is an important read for everyone thinking about iPhone apps." —**Jürgen Schweizer, founder of Cultured Code**

Tapworthy

Designing Great iPhone Apps



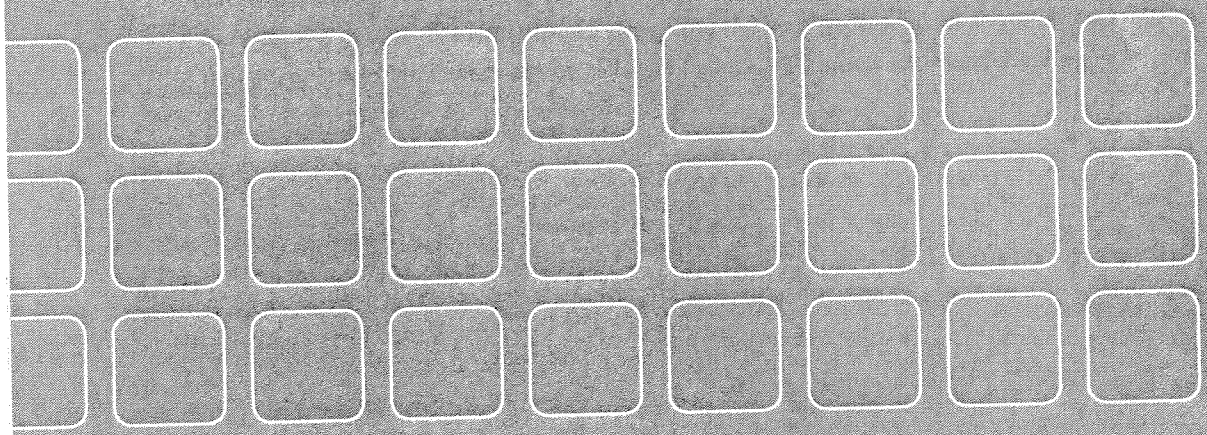
O'REILLY®

Josh Clark

2

Is It Tapworthy?

CRAFTING YOUR APP'S MISSION



IS IT WORTH IT? That's the calculation running in your users' heads with every tap and swipe. Just by launching your app, users have to spend scarce resources—time, attention, thought—that are in especially short supply for mobile apps. What do they get in return? You just saw how mobile users churn through apps at the speed of distraction. Unless you meet their needs and, even better, entice them to slow down and explore, they'll keep on going. Tapworthy design starts with a firm understanding of your audience and their goals.

In the big picture, an app is tapworthy if it makes your users' lives better by helping them get stuff done, make them laugh, stay connected, fill downtime, or do whatever they otherwise need to do to be awesome in that moment. Tapworthy apps might be easy on the eyes, too, but the fundamentals of great design don't hinge on making things pretty. In app design, beauty derives from function, and every interface element has to be focused on helping your users do what they're there to do.

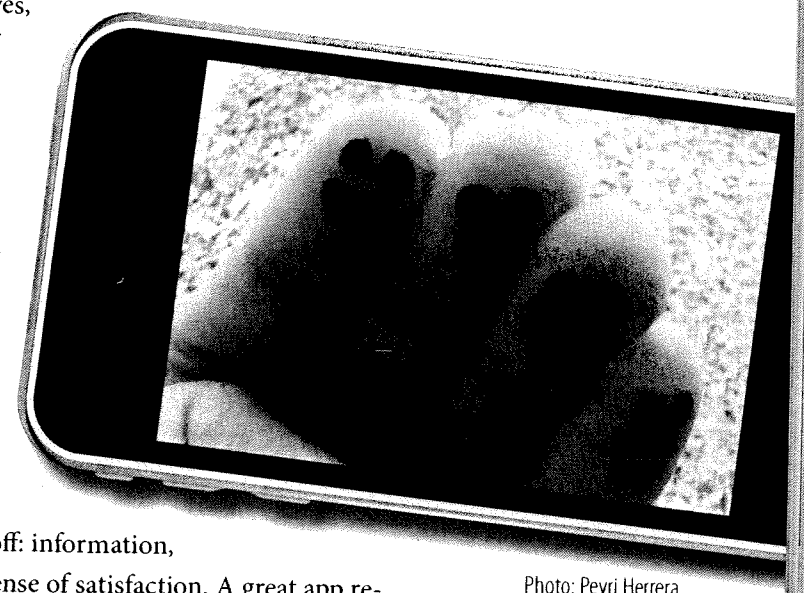


Photo: Peyri Herrera

Designing tapworthy iPhone apps means designing for an economy of time, attention, and screen space. Every tap should have a pay off: information, delight, a completed task, a sense of satisfaction. A great app rewards the user at every turn, from the first glimpse of its app icon through every tap and swipe. This takes both careful editing and definition of purpose. Clearly stating what your app does and how it's unique brings needed

focus to the design process. You'll start to dig into the details of designing for the small screen in the next chapter, but before you start slinging pixels and making interface decisions, you have to start with more fundamental choices: *What does your app do . . . and why?*

There's Not an App for That

If only fresh ideas meant automatic success. "Build a better mousetrap," the saying goes, "and the world will beat a path to your door." Lots of would-be mousetrap millionaires have taken that advice to heart: Over 4,000 patents for mousetrap designs are on file in the US, but only about 20 ever turned into successful commercial products. The dense thicket of apps in the App Store is an even more concentrated example, with the vast majority—even worthy ones—languishing in obscurity and indifference. There are lots of reasons for an app to flop, but it doesn't help if the problem was already solved by another app . . . or perhaps the problem never needed solving in the first place. Our friends in the mousetrap industry learned the hard way that it's tough to improve on the no-frills snap trap; better to invest your efforts in something altogether different, something new and needed.

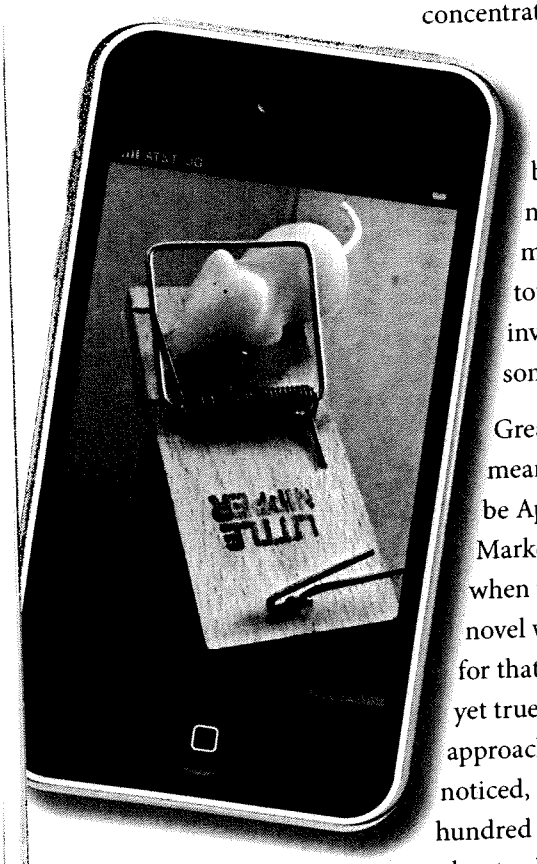


Photo: Angus Stewart

Great design is a worthy pursuit in itself, and I don't mean to suggest that your goal as an app designer must be App Store success, whatever that might mean to you. Marketing and design considerations do align, however, when you meet your audience's needs in an effective and novel way. If Apple's marketing mantra is "There's an app for that," make it your goal to find a case where that's not yet true. With the number of apps in the App Store swiftly approaching the gajillion mark, it's not easy to get a new app noticed, and you won't help matters by mimicking what a few hundred other apps are already doing. If you're building yet another to-do list app, tip calculator, or flashlight, be sure it does

something different from (and hopefully better than) the throngs of similar apps that have already found cozy homes in the App Store.

This is Marketing 101, sure, but it leads to a crucial question: *what specific problem does your app uniquely solve for users?* Too often, people start from the other end of the stick, effectively asking, "What does this app do for *me,* the app creator?" Maybe there's an iPhone feature you're itchin' to work with, or your company has specific content it wants to get out there, or you have astounding skills in a particular technology. It makes good sense to build on your passions and strengths, of course, and those considerations are sensible ways to choose the broad domain for your app. But that addresses only what you (or your company) will get out of the app, not users. You have to bend your content, interests, and competencies to meet bonafide user needs.

Features, content, and gee-whiz animations may be crucial building blocks for your app, but they're not the *reason* to use your app. At the broadest level, it's the reason—the *why*—that makes an app tapworthy. People will use your app if it solves a problem, gives them a superpower, or just helps them unwind, but without a clear, persuasive vision of when and why people will use your app, you're just building a technology demonstration, a curiosity.

What's Your Story?

The best apps give users elegant solutions to precise needs—the more focused, the better. As you plan your app, think in terms of actual *use cases* or *scenarios*—brief story lines that cast the user as hero completing specific tasks in specific contexts. Like any good yarn, your app's story should answer "the five W's" that budding news reporters learn to pack into their lead paragraphs: who, what, when, where, and why. "Who" identifies the audience for your app, "what" identifies the actions they'll take, "when" and "where" zero in on context, and "why" describes their motivation and goals. By focusing on these story elements of your use cases—especially the *why*—you'll uncover your app's tapworthy conditions, the moments your users will need your app's superpowers.

Here's the hitch: in order for this to work, your use cases have to be plausible. There has to be an audience of people who not only want to do what you describe

(the "who" and the "what") but more important, *they have to want to do it on their iPhone* (the "where" and the "when"). It's all too easy to lose sight of this mobile context. The iPhone is, after all, a full-fledged computer with an Internet connection and a grown-up processor capable of all kinds of complex tasks. For folks used to designing websites or desktop software, it's tempting to think of the iPhone as a "regular," if downsized, computer. While that's true in a technical sense, it doesn't translate to real-life use.

People behave very differently when using mobile apps on the go than we do when typing away at our desks. Just because you can put sophisticated software or a complete content reference on an iPhone doesn't mean anyone will actually want to use it there. The iPhone's form factor and relatively underpowered processor mean that the device is better for some uses than others. That means the "why" of your use cases has to embrace not only why users would use your app's features or content but why they'd use it on an itty-bitty handheld device.

What Makes Your App Mobile?

As extraordinary as your app might be in features, content, and technical razzle dazzle, it's only tapworthy if your users find it convenient, necessary, and easy to use in a mobile context. "Mobile" means on the go, of course, but in the iPhone context it's helpful to think of its meaning more flexibly as "away from my desk." Whether you're on the peak of Kilimanjaro or just curled up on your couch, both are mobile contexts—each with their own opportunities and potential distractions. What mobile context are you designing for? Why would you use this app when you're away from your desk or computer? Why is it especially convenient to have anytime-anywhere access to this app in your pocket?

Sometimes this is a no-brainer; some apps are naturally mobile because their whole purpose is to be used in the field. Take a gander at iBird Explorer Plus, a sprawling field-guide encyclopedia of birds and bird calls. It's an app for bird watchers (*who*) to look up info and birdsongs (*what*) when peeping at their fine-feathered friends in the wild (*when* and *where*) to identify a bird or attract one with a bird call (*why*). This is a niche audience, to be sure, but it's also a natural mobile app whose value is undeniably tapworthy if you happen to be a birdwatcher

in the brush. By wrapping your five W's tightly around a mobile context, you've got the makings of a must-have app for your audience.

iBird Explorer Plus, like any field guide, is intended for the field. Its utility as a mobile app is naturally baked into its essential concept.

iBird is an example of an *accessory,* an app that augments an activity—a bird-watching expedition in this case—but accessories don't have to be so *explicitly* mobile. Other iPhone accessories like a calculator, guitar tuner, or recipe collection are just as useful on your couch or in your kitchen. These, too, are mobile contexts—nontraditional computing environments— where you can craft a convincing set of five W's for an app to extend and enhance another activity. No matter what the specific setting, consider how your app can take advantage of the size and portability of the iPhone to do something that desktop computers cannot.

TAKE ADVANTAGE OF FORM FACTOR

Mobile Mindsets

As you ponder the features and mission of your app, always return to the mobile use cases where your app might provide a solution. A good way to do that is to step back and think about the reasons we launch apps in the first place. Of course, there are as many individual reasons to use the iPhone as there are iPhone users, but it turns out every mobile impulse typically boils down to one of three mindsets:

- "I'm microtasking."
- "I'm local."
- "I'm bored."

"I'm Microtasking"

As smart phones keep getting smarter, more and more of us are leaving our laptops behind, leaning instead on our trusty phones to keep up when we're away from home or office. Unlike laptops, notebooks, or messy collections of Post-It notes, our iPhones rarely leave our sides, making them handy vessels for bottling brainstorms, managing to-dos and itineraries, or capturing on-the-go information like expenses or billable hours.

Slowly but surely, we're learning to get stuff done on the small screen, but it's important to keep this miniature revolution in perspective: the iPhone can't match a laptop in many respects, and you shouldn't assume people will use it the same way. No matter how blazingly fast your fingers and thumbs might be, for example, delicately pecking away at a touchscreen keyboard won't win any speed records. The iPhone is better tuned for parking quick notes than powering through the Great American Novel. A new style of device encourages a new set of work habits.

The iPhone is a device of convenience and context, ideal for short dashes of activity—*microtasks*. The best iPhone apps emphasize quick access to ideas, contacts, tasks, info, forecasts, or entertainment. These apps thrive on simplicity, offering fewer features than desktop counterparts, but also making it faster and easier to get it down quick. Compared to traditional computer work of long, sustained

work sessions, the best *productivity apps* are tuned for short but frequent hits, encouraging users to capture new information and ideas as they happen, typically to be processed and massaged later. The iPhone is likewise ideal for reading and even editing documents in the otherwise lost time of grocery-store lines or subway commutes. (This also happens to be true of the best iPhone games, which are typically designed for quick but tasty bites of gameplay, just a few minutes at a time.) There's room for more leisurely exploration, too, as you'll see on page 37, but in all of the iPhone's contexts, the device's quick-draw convenience lets users make the most of downtime, whether for work, play, or creative contemplation.

Productivity and reference apps in particular should be tuned to make the most of these intrawork interludes, making themselves tapworthy with efficient interfaces that are well-suited to this evolving style of work. As you consider your app's use cases, build the resulting features around microtask sprints of brief activity. Identify the recurring tasks that your users will perform with your app, and then polish, polish, polish. Optimize the design and workflow to make those tasks quick and effortless to accomplish on the go. Tapworthy apps get it done fast.

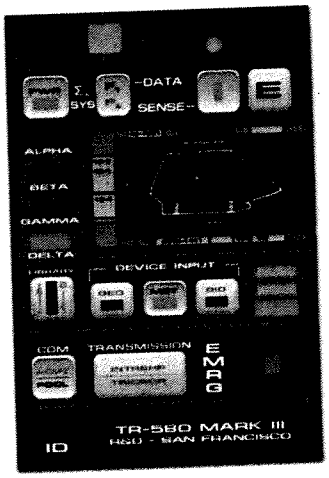
Tapworthy apps accommodate users in a hurry, optimizing for frequent, recurring jobs. The to-do list app Things (left) makes it fast to add new tasks from any screen: Just tap the + icon that's always parked at the bottom left of every screen. The built-in Calendar app takes a similar approach for new events, placing its + icon at the top right of all screens.

2 "I'm Local"

We launch mobile apps to get the skinny on our surroundings: "I'm local, tell me what's happening around me." We've celebrated the personal computer since

the days of disco, but the iPhone is the most personal computer yet—a device that knows tons about you and your surroundings. More than just a precocious phone, the iPhone is a personal sensor device tricked out with a camera, microphone, GPS, motion detector, and compass, all backed with Internet know-how. With sight, hearing, and touch, it lacks only smell and taste to round out the five senses (and really, that would just be creepy).

The iPhone is at once hyperlocal weatherman, restaurant radar, source of directions, know-it-all travel guide, and more. Trekkies, eat your heart out. Sensor-savvy apps turn the iPhone into the tricorder of Star Trek yore. Like a pack of pointy-eared science officers, we whip out our phones for a confident read on our personal habitat, using the iPhone's sensors to filter sprawling amounts of information for a local view of our immediate environment. Fascinating.



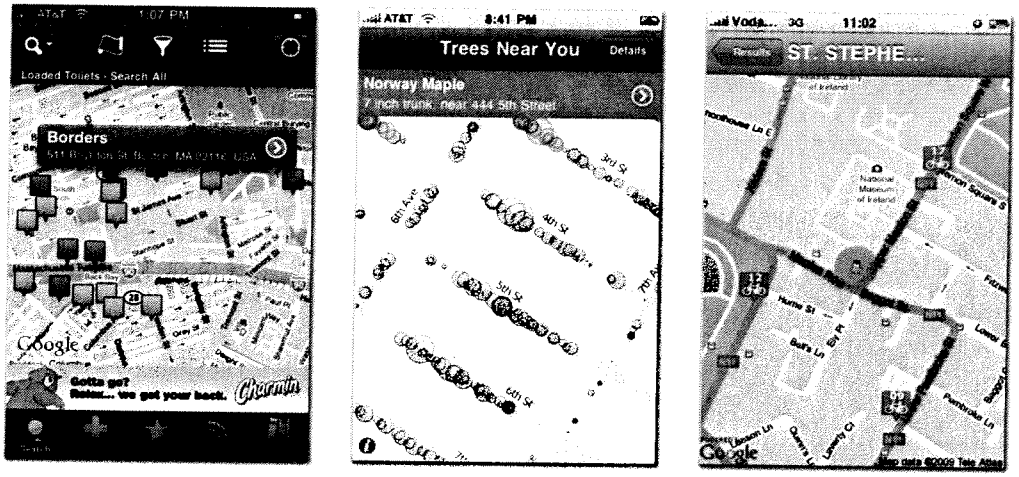
Get your geek on. The Tricorder TR-580 app lets Star Trek devotees fire up an ersatz tricorder. As packed with sensors as the iPhone already is, though, it turns out this novelty app isn't so far from reality.

Tapworthy apps take advantage of iPhone sensors to give personal context to tasks and info where appropriate. For most location-based apps, the whole goal is to put an appealingly nearsighted lens on a vast universe of data. The built-in Maps app is of course the most familiar example of this type of app, and the App Store is full of plenty of novel map-driven apps. A few examples among many:

- Yelp lists nearby businesses and dishes reviews from über-opinionated locals.
- Zillow is a drive-by home shopper's dream reference, mapping nearby homes for sale, along with property values in the neighborhood.
- HearPlanet is an on-the-fly audio guide, speaking descriptions of sites and landmarks around you.

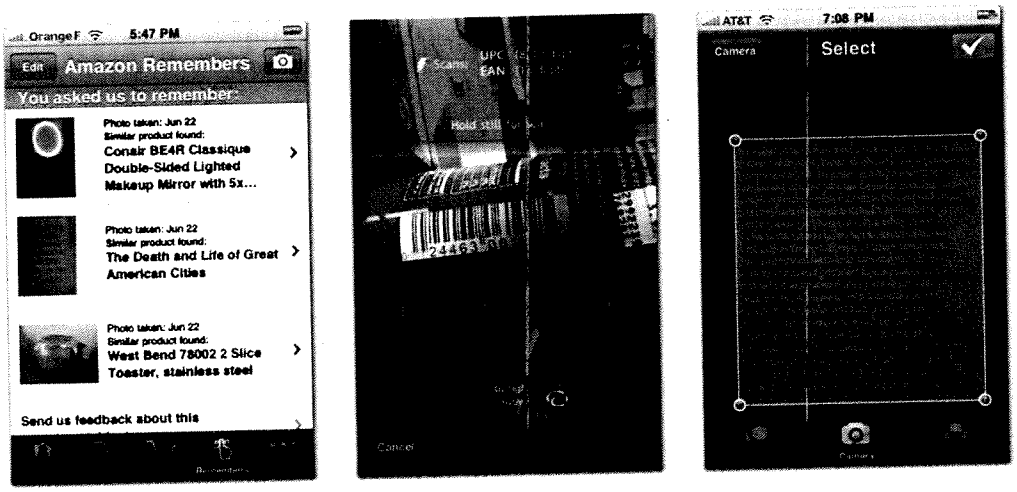
As useful as these what's-nearby apps can be, the genre has quickly grown familiar—perhaps over-familiar—as general-purpose local guides have popped up faster than you can say, "Where's the nearest Starbucks?" With any app, it's important

to stake out how it's different from the rest, but if you're playing in this crowded sandbox, it's especially crucial to do something beyond what the other kids are doing. As usual, paying special attention to defining a unique set of five W's helps you stake out your own tapworthy territory on the map. Think hard about how you can craft your location-based app around a very specific audience (*who*), content (*what*), or need (*when* and *why*). The highly personal context provided by the iPhone's location sensors encourages a peculiarly personal niche focus. As more and more geotagged data makes its way into the world, we're surrounded by data ghosts whispering information about our immediate environment. A creatively tuned iPhone app is like a set of special goggles for bringing *some* of these ghosts into view, letting your audience focus on a hobby interest or particular need.

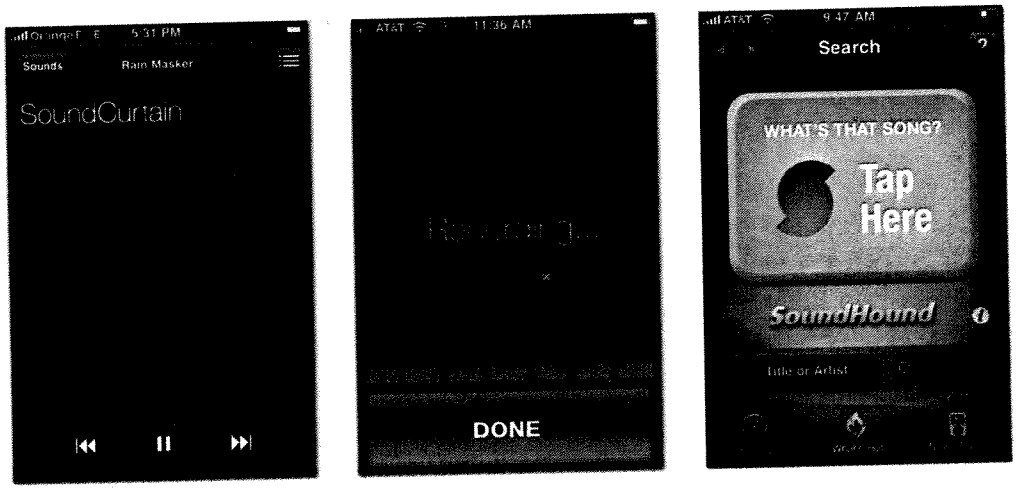


Identify your audience, and help people connect with their passions (or maybe just a restroom at an urgent moment). These apps make themselves unique with narrow content and focused audiences. From left: SitOrSquat finds nearby public restrooms; Trees Near You tells you what type of trees are around you if you happen to be in New York; abikenow serves the public bike-sharing programs of Dublin, Brussels, and Lyon, telling residents where bikes are available for pickup and spaces available for drop-off.

Location-based map mashups are perhaps the most obvious way to put iPhone sensors to work, but it's worth thinking creatively as you plan your app about other ways that personal location can be put to use. The camera and microphone, for example, let you go beyond "what's nearby" to provide info about "what's in front of me." Audio-minded apps can analyze your sound environment for dictation, transcription, making music, or even helping you tune out extra noise.



Shutterbug apps use the iPhone's camera to dish info about what users are looking at. Take a photo of just about any product with Amazon Mobile (left), and the app will identify it and provide an Amazon link for more info. RedLaser (middle) does a similar trick, but with bar codes; aim the camera at a bar code, and the app tells you where to find the best price. Babelshot (right) translates photographed text to and from scores of languages, a nifty trick for travel.



Sensor-savvy apps can listen in on the microphone to provide feedback and stow info. SoundCurtain (left) shuts out noise by playing ambient sound through the headphones, adjusting the volume based on the sound around you. Dragon Dictation (middle) listens to your voice to transcribe brief snippets of speech. Midori Soundhound (right) identifies songs playing in the background, even ones that you hum to it.

These examples feature the iPhone sensors as the main event, tied directly to the apps' key features. But there are also opportunities to provide more subtle interventions, providing location-based info, for example, in apps where geography hasn't been a traditional concern. In to-do list manager OmniFocus, for instance, you can use your location to show tasks that are tied to nearby locations, handy for tackling errands while you're out and about. Similarly, Shopper is a grocery list organizer that detects what store you're in and organizes your shopping list according to the order in which you'll encounter the aisles and departments of that store.

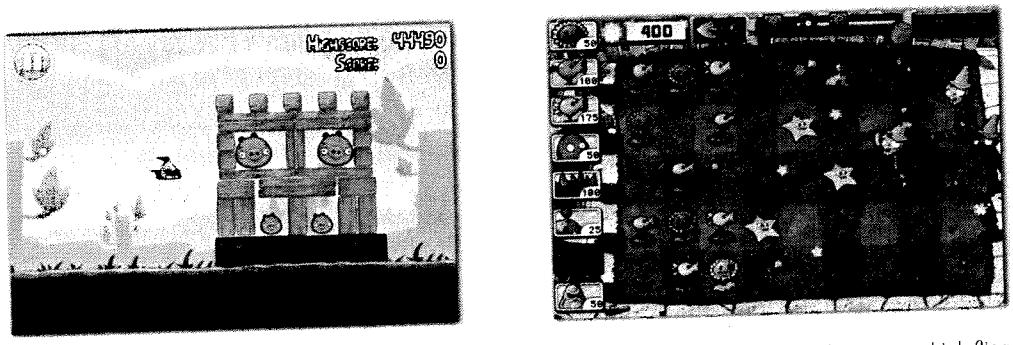
When you put a fistful of sensors in users' hands, it opens lots of opportunities to present tasks and information in entirely new ways. Tapworthy apps consider what it means to be local and act on that knowledge to change the app's interaction. Don't force it, though; think practical, not gimmick. Not all apps need to be photo-fantastic, audio-optimized, and geographically generous. But used in the right place, the iPhone's sensors can personalize your app to make its features more helpful and relevant.

"I'm Bored"

Let's not kid ourselves. All this talk of microtasking and local data analysis makes it sound like every iPhone-wielding citizen is a paragon of productivity. The iPhone is swell at getting stuff done, but your glossy gadget is even better at wasting time. For every productivity enhancer in the App Store, there are at least three productivity killers, and that's a good thing. The world shouldn't always rush rush rush, and well-crafted iPhone apps can help us unwind, relax, and find shelter for mindful moments. The iPhone is great for staving off boredom for the same reason it's great for microtasking: it's always with you, at the ready to fill downtime with easy distraction, giddy gaming, or even high-minded escape into the world of literature. What could be more tapworthy than helping you survive a dull-as-paste moment?

Even a quick peek at the App Store's numbers reveals that Apple's online emporium is more arcade than office. Games dominate, accounting for three quarters of the most popular paid downloads since the store opened. And it's no wonder: the

iPhone is a fun, quirky, and genuinely delightful gaming device. Its limited but intuitive controls—the touchscreen and motion-detecting accelerometer—make iPhone games accessible to everyone, but fresh enough to grab seasoned gamers. With a huge library of games, most available for less than a pack of Slim Jims, there's no shortage of casual games to enjoy in bites of a few minutes at a time. We're hooked.



Casual games are ideal boredom busters. As shown here, games let you escape into worlds where cartoon birds fling themselves at an occupying army of pigs (Angry Birds, left) and where lawn plants defend against a horde of invading undead (Plants vs Zombies, right).

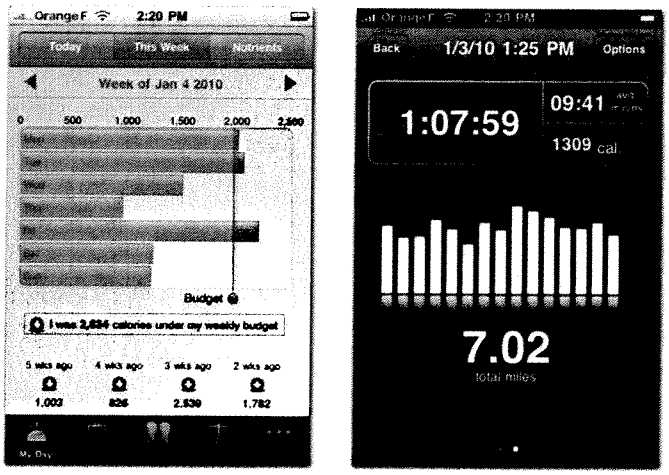
The boredom battle doesn't fall only to guns-a-blazin' video games. The antidote for boredom is simple enough: something that's better than what I'm stuck with right now. When we're bored, we want something to occupy us beyond the bleak reality of the Post Office line, the tedious meeting, the eternal wait at the bus stop. *We want something to do.* And, often, we want something to make us laugh—a fact that has, on mobile devices, shifted the whole nature of consumer software. It turns out that *millions* of people think that novelty apps for making fart sounds are a gas. The commercial appeal of toilet humor isn't anything new (just ask Chaucer), but it's the first time that it's become a full-fledged software genre. Until now, most folks bought computer software to do work, period. Now, on the iPhone, users want entertainment, too, even if it's occasionally vacuous or flatulent. On this device, people see software as content, not merely a set of tools.

This doesn't mean your app has to rumble with intestinal distress. But it does mean that you should consider how your app might give your audience some moments of delight and distraction, or even encourage them to slow down and lose themselves for a while. All but the most simple utility apps (like the built-in

Weather and Clock apps, for example), can give users something to explore or play with to pleasantly pass the time.

The common thread to boredom-busting apps is *exploration*. A great app gives you someplace to go, a world to creatively travel or distractedly get lost. That's why video games are such effective crusaders in the war on boredom, and why ebooks, news apps, YouTube, and Twitter clients are so popular on the iPhone, too. They all provide a story, an escape. But what might not be so obvious at first blush is that relatively mundane, workaday apps can provide a similar experience of exploration, too. Even get-it-done-quick apps can afford opportunities to slow down for thoughtful contemplation.

That's especially true for apps that collect personal information. Calorie trackers, fitness logs, and to-do lists, for example, are essentially catalogs of the user's past or future achievement. When presented in the right way, these apps transform themselves into personal-history video games. While these apps are (and should be) focused on micro-tasks for collecting and tracking info, an important use case is to allow users to massage that meticulously gathered data to see their progress and sort out where they're headed.



Health and fitness apps, like all apps that collect user info, gives users boredom-busting opportunities to explore their personal histories. Lose It! (left) lets you browse your calorie intake on weekly, daily, and meal-by-meal basis to review your weight-loss progress; here, a chart lets you explore your daily calories. RunKeeper (right) stores your running stats, letting you revel in minute-by-minute reviews of all of your runs. Here, a bar chart shows your pace for every minute of the run; tap a bar to see its details.

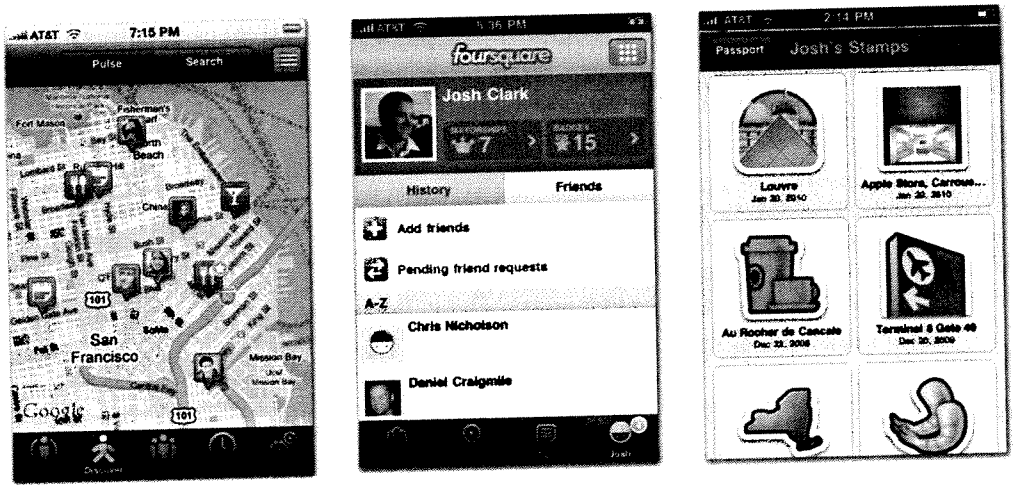
As important as it is to plan your app to accommodate frequent, rapid micro-tasking, don't forget the more occasional but equally important leisurely crawl through the app. Whether it's the user's own content, usage history, fresh news,

community, or recommended content, consider what you can give users to explore. That's your boredom buster.

What Makes You So Special Anyway?

So you've got an idea of what you want your app to do, and you've got a pretty good sense of how and why people will use your app on the go when they're multitasking, local, or bored. You're off to a good start. But chances are, someone's already got an app that does something at least vaguely similar to what you've got in mind. How will your app be different? What will make your app stand out from the rest?

With a new angle, even upstart newcomers can get a toehold in territory already staked out by established apps. Consider the category of "check-in" apps that let you share your current location with friends. Loopt was among the first iPhone apps to do this, providing an efficient system for plotting your pals on a map, making it easy to meet up with them wherever they might roam. As more apps joined the category, the best of these fresh faces added a new spin that made check-ins more fun. Foursquare added a point system to the formula, turning the activity into an urban game where everyone in the city is a player: check in



From left, Loopt, Foursquare, and Gowalla all offer similar functionality, but each offers its own spin, providing three very different experiences and incentives for sharing your location with friends.

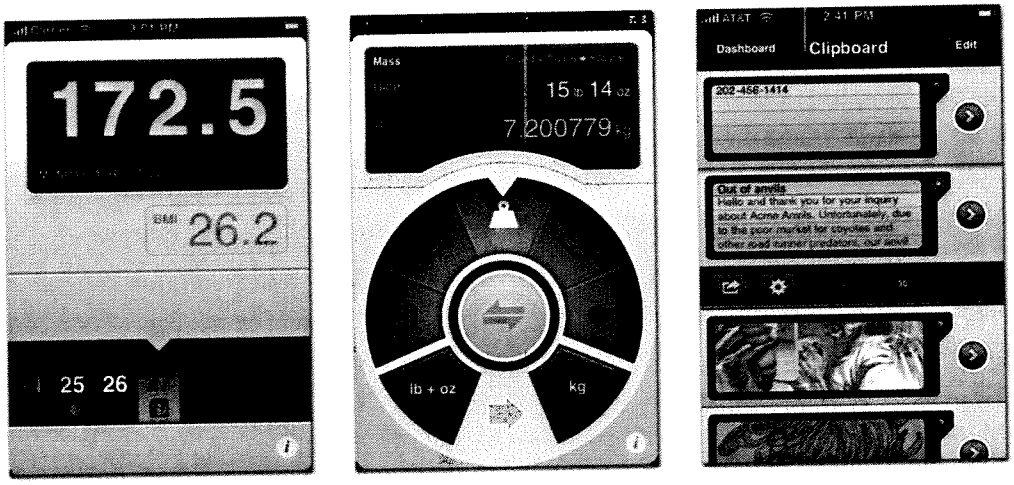
more than anyone else at your favorite watering hole, and you become its mayor, a competitive incentive for exploring your own city. As you saw earlier in this chapter, Gowalla tweaked the recipe by letting you collect virtual objects, pick up icons, and add "stamps" to the app's passport as you roam your city. Instead of scoring points, Gowalla focuses on serendipity and the surprisingly addictive discovery of unexpected objects when you go to a new spot.

All three of these apps focus on the same activity—announcing your location—but all offer distinctly different rewards for doing it. These different approaches—efficiency, competition, and discovery—give these apps distinct personalities which in turn suggest different audiences and use cases. Despite surface similarity, the three apps are unique.

What makes *your* app special? Some possibilities: a unique set of rewards or incentives; a tight focus on a specific audience; niche content that no one else can provide; a new way to visualize or present information; technology that simply works better; a big network of other users to play with; a solution for saving money over other apps; or a website or real-world component that enhances what you do or see in the app.

All of the advantages in that list are, in a sense, "skills." If you do any of those things better or differently from others, you have a hook. But it also helps to think about apps like we think about human beings. It's not just skills that make us want to spend time with certain people. There's also the slippery but irresistible matter of charisma. Personality and looks matter, too (sometimes more than we care to admit), and the same goes for apps. Much of this book emphasizes efficiency, focus, and substance, but there's no doubt that style matters. Even if your app does exactly the same job as everyone else, you make an impact if it does that job with a flourish.

Just go carefully. Like people, apps with oversized personalities are as likely to distract and disgruntle as they are to seduce. Overdoing it with animations and sound effects will irritate users who don't want to be bothered. If you choose to make your app's graphical style one of its main differentiators, be sure that the style isn't so noisy that it drowns out your app's actual info and features. Adjust the style and design to suit the content and audience, and be careful not to



Tapbots is an iPhone development shop whose apps push hard on personality. The company's apps do tasks that are mundane to the point of blandness—a weight log, a unit-conversion calculator, and a clipboard manager—but their clever interfaces are frankly adorable. Shown here from left to right, Weightbot, Convertbot, and Pastebot all feature retro-robotic designs complete with whooshing hydraulic sounds that make it, yes, *fun* to convert miles to kilometers. Design and whimsy are features, too.

become so smitten with your app's good looks that you begin to confuse form with function. Great design isn't just about aesthetics.

Wait, Wait, Come Back!

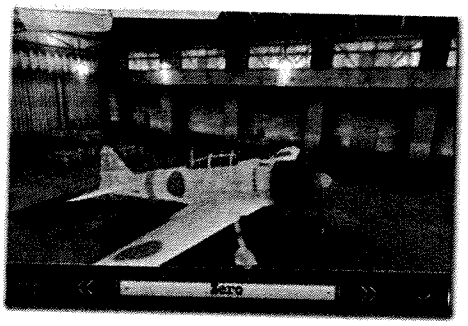
Your app's shelf life on a user's Home screen is exactly as long as it can hold that user's attention. iPhone owners chew through apps, gulping down their content, then tossing them out and moving on. Apps with a fixed amount of content or data are particularly vulnerable to getting jilted; once your users read your ebook, play all the levels of your game, or flip through all the flash cards of your vocabulary builder, there's nothing left to keep them around. If that's your intent, that's fine; some apps simply have a limited lifespan, and once the content is exhausted, that's the end. But if you're trying to create a long-term relationship with your audience, your app has to keep giving. It has to have a heartbeat to stay alive and remain tapworthy.

Certain kinds of apps have a built-in heartbeat thanks to the fundamental nature of their key features. Tools that organize personal info (to-do lists, calendars,

expense trackers, contacts) keep people coming back as often as they need to check an appointment time or add a reminder to pick up the milk. Likewise, utilities that perform a common task (Skype calls, instant messaging, barcode scanning, weather forecasts, notebooks) continue to be useful as long as the underlying task itself is in demand.

For most of us, tools and utilities still account for the majority of our desktop software. Most folks use computers to work . . . to *do*. That doesn't hold so true for iPhone apps. Instead of tools, the majority of apps in the App Store are some flavor of *content app*—games, entertainment, books, references, novelty apps. When we use this category of apps, we're consumers, not doers. When there's nothing left to consume, we move along to a new app. For these apps, it takes more work to keep the heartbeat thumping. Sometimes that simply means fresh content. News apps, of course, have a bottomless and constantly refreshing reservoir of news, drawing users back regularly for more about the latest political brouhaha or Brangelina update. For these apps, as long as the content continues to appeal, users keep coming back.

Non-news content apps have to be a bit more creative, but the challenge is the same: to continue to provide fresh content. Games can offer additional levels through in-app purchases, which also has the happy side effect of steering additional cash into the company checking account. But new game levels are about more than just an opportunity to keep playing; they're markers of achievement.



In-app purchases and add-ons let users keep the good times rolling . . . literally, in the case of the Skee-Ball-themed game Ramp Champ (left), which lets you download new ramp themes for free or fee. Skies of Glory, an aerial combat game (above), lets players upgrade the game's gear for a fee. Here, a Zero plane will run you three bucks.

STEADY UNLOCKING OF CONTENT

You've beaten the game and you're ready for more, even ready to *pay* for more. There's a collect-em-all mindset, and the game grows as you get better. That approach might seem unique to games, but other types of apps can offer a similar sense of expansion as achievement, letting you unlock new content as you master various chapters, challenges, or lessons.

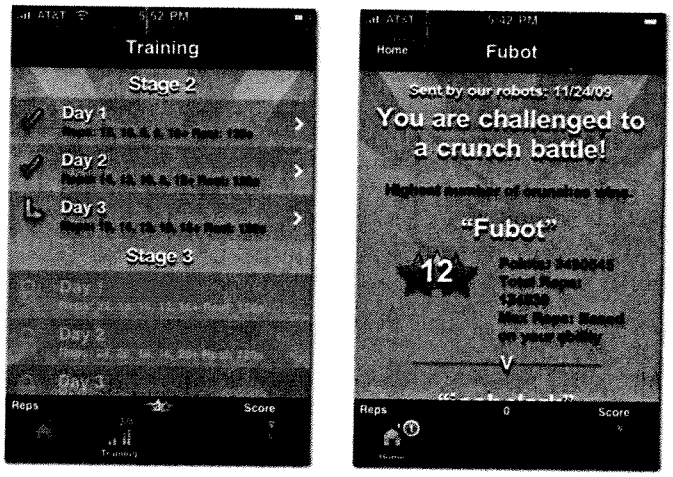
Travel apps are especially road-ready examples of apps that benefit from content that expands and adapts to mirror users' activities. Lonely Planet Travel Guide, for example, comes with a single city guide to San Francisco, but offers scores of other travel guides and phrase books for purchase inside the app—buy a new guide when you're headed to a new city. The app's content collection expands according to your needs (and pocketbook), so it stays as fresh as your arrival in a new city. This collection of destination guides meanwhile memorializes your globe-trotting meanderings as readily as stamps in your passport—or vanquished levels in a game. Similarly, OffMaps lets you download maps and guides for offline use anywhere in the world, handy for dodging international roaming fees that otherwise make it ruinously expensive to use the built-in Maps app abroad.

Travel apps can expand their content as you expand your travel horizons. OffMaps (left) lets you download maps as you need them. Lonely Planet Travel Guide (right) offers new guides for sale inside the app, ready when you arrive in a new city.

This steady unlocking of content is also a staple of fitness apps. CrunchFu, for example, is an app for acolytes of the six-pack ab, a training program to help transform bellies from wishy-washy to washboard. The app gives you a daily training program to do a recommended number of crunches. As you complete each day, you unlock the next day's program, each one gradually more difficult until you

finally gut through 200 crunches at a single go. These steady, game-like accomplishments keep users coming back while also encouraging a safe fitness ramp-up, an effective way to give the app a healthy heartbeat.

CrunchFu keeps users coming back with its progression of gut-busting training programs. Even when the program is done, though, the app's head-to-head "crunch battles" maintain interest.



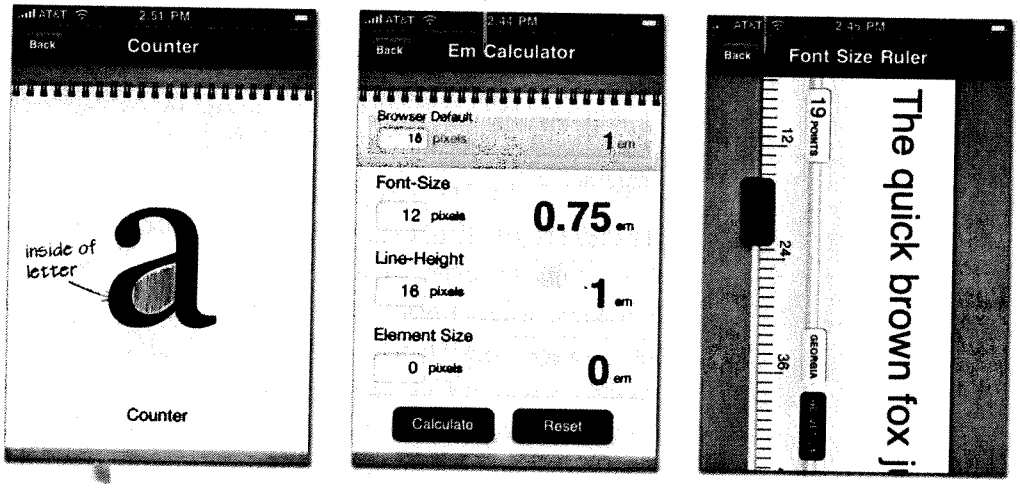
Even so, CrunchFu's content is limited by the fact that its training program has a finish line. After you hit 200 crunches, you're done. Normally, when users finish an app's content, they'd amble off into the sunset, never to look back at the app again. Once again, though, gameplay saves the day. Even when the training program peters out, CrunchFu provides a game to hang on to some portion of its users. The app offers "battles," head-to-head crunch competitions with other CrunchFu users. No matter where you are in the training program—beginning, finished, or somewhere in between—the app lets you find someone else at your fitness level and challenge them to see who can do the most crunches, earning points as you go. Human contact for the win!

It might not look like it as we tap away at our iPhones, off to ourselves and oblivious to the world, but we are irresistibly social creatures. iPhone or not, we're drawn to activities that let us communicate, compete, or contribute. Like a physical place, apps that bustle with the activities of other people feel alive. Community features give an app life beyond its fixed set of content. Even after you've mastered a game, the chance to test your skills against other players gives you an incentive to keep playing. For other apps, sharing content with friends and seeing what others are saying lets your users provide an ever-replenishing supply

of fresh content. Among its many other features, for example, the Movies app by Flixster pools reviews by regular-Joe moviegoers. You can follow the advice of the grand mass of public reviews, or just listen in on what your friends have to say. Yelp adds similar value by encouraging customer reviews for local businesses, and Amazon, of course, does the same with reviews of its products. In all of these cases, community-driven reviews give extended life to the basic content provided by the app developers.

You don't have to build this stuff yourself; you don't even need your own community. Just plug your app into the established social networks, where millions of people are already talking. Facebook and Twitter both provide easy platforms for sharing content from iPhone apps. For games, social networks like OpenFeint or Apple's Game Center let developers plug high-score leader boards and head-to-head challenges into their apps.

The idea is to wrap secondary features around your app's main content to enhance and extend its value. The apps described so far have done this by growing or sharing that primary content. But another effective, if less elegant, approach is simply to bolt on complementary tools. The Typography Manual, for example, is an app for budding graphic designers. It's essentially an ebook spelling out the history of typography, tracing its letterforms and laying out its technique. While the app does a fine job of dotting the i's and crossing the t's of its subject, it remains like any book, a fixed set of words and images. But the app also throws in a handful of tools that are useful to folks who sling type for a living. There's a font ruler, some font-size calculators, and a comprehensive reference for easy-to-forget HTML codes for special characters. Even after users finish reading the book, these add-on utilities keep them coming back by continuing to provide some modest value.



In addition to its reference content—a visual typography glossary (left), an ebook, and an HTML reference—The Typography Manual includes a font-size calculator (middle) and a font ruler (right) to provide lasting utility even after you've consumed the app's content.

As with all things iPhone, however, use restraint. Think hard about the features that can give your app extended longevity, but don't just pad the app with features willy-nilly. It's often a useful service to your audience to design your app for a steady, long-lasting heartbeat. But as you do this, it's even more important to keep the app simple and focused.

Throw Out the Babies, Too

The initial planning for your app should be big and bold. Be expansive in how you think about its features, the audience it should address, the opportunities to set the app apart from the rest. We're still in the early days of exploring the possibilities of devices like the iPhone, and invention is important. Let the crazy ideas roll. Fill the drawing board with rainbows and unicorns, and drum up any feature you can think of that will make your app do things no app has ever done before.

And then, take the features on your lovingly crafted wish list and throw most of them out.

"Murder your darlings," advises the favorite writers' adage, counseling flowery wordsmiths to be ruthless about trimming flabby text. "Whenever you feel an

impulse to perpetrate a piece of exceptionally fine writing," said the wag behind the homicidal phrase, Sir Arthur Quiller-Couch, "obey it—wholeheartedly—and delete it before sending your manuscript to press." Explore the possibilities, in other words, but remember that an economy of well-chosen words does more for readers than literary derring do.

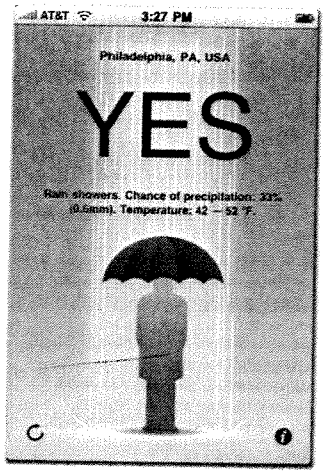
So it goes with mobile apps. An effortless user experience requires a streamlined selection of tapworthy features. It all goes back to the iPhone's environment of scarcity—limited attention, time, pixels, device memory, and processing power. Every feature you add has to compete for these scarce resources. Add too many features, and the app experience bogs down as surely as the purple prose that exasperated Sir Arthur. The interface grows cluttered, the workflow slows, and potentially tapworthy features become obscured. Instead, your app should be tightly focused on a unique task and a handful of use cases. While you don't have to throw out all of your rainbows and unicorns, you should spare only the small handful your users will need to get the job done. *Think big but build small.*

To start, identify the single task that's most important to your app. Go back to the most important use case you've crafted for your app—the who, what, when, where, and why—and identify the recurring task your users will most frequently need to tackle to knock out that use case. If your app is a to-do list, the key task might be adding a new to-do item. If it's a calendar, it might be scanning the day's events. All of the design work you do from here will flow from this decision. Once you've identified the key activity, think hard about what your app can do to help your users microtask that activity in a hurry.

Secondary tasks will naturally emerge from that primary activity. Once you can add items to your to-do list app, for example, you need to be able to review, edit, and check them off, too. Keep it simple, though. Figure out the bare minimum required to bring your idea to life, then optimize the experience with delight, efficiency, and polish.

The app doesn't have to be crammed with features and functions to be the best in its category. In fact, a lean diet of features often delivers a better user experience that's more carefully tailored to specific needs and use cases. You might even pare the entire app down to answer a single important question for your audience.

"Umbrella: The Simplest Weather Forecast" is a disarmingly simple app approaches weather forecasts by answering the question, "Do I need an umbrella today?"

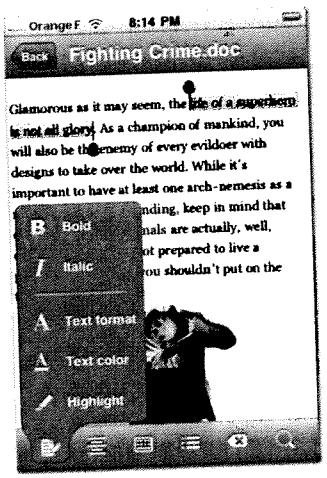


The less-is-more approach probably doesn't fit your gut instinct about how to build a great app. Developers of websites and traditional desktop software are accustomed to waging a features-arms race with competitors. Enterprise software outfits evangelize the idea that the more features you have, the better the software. Marketing for new versions trumpets all the new-fangled gizmos that have been tacked onto the update. The benefits of feature pile-on are dubious on the desktop—the result is often bloated software—but the effects of an overstuffed iPhone app are downright hobbling. Don't assume that your app has to do *everything* related to your content domain. Pick your slice of targeted work to do and let other apps pick up other slices. (In Chapter 11, you'll learn how you can make your app fit into this larger iPhone ecosystem.)

Of course, the trick is picking the *right* slice, the appropriate mix of features for your selected audience. As you winnow features, merrily murdering darlings along the way, stay tuned to the needs of the majority of your users. Choose the features that *most* of your users will need most often, and don't fret over the esoteric needs of a minority fringe. That's common sense, sure, but the fringe is often the noisiest, the power users who gripe in iTunes reviews that your alarm-clock app is a bust if it doesn't display seconds in atomic clock units. *No caesium-133 radiation cycles? FAIL. I want my 99 cents back!* Simmer down, Marie Curie. There's nothing wrong with caesium-133 radiation cycles, but of course they have no meaning to a general audience. That said, the fringe can sometimes tip you off to a potential niche app. Who knows, there may very well be a tidy market for an atomic alarm clock app—in which case, the fringe *is* your audience, and mainstream users become the fringe. Form a clear picture of your audience and tailor your app's features to the majority of those users.

At the same time, in all this paring down, be sure that you're still giving people at least a minimum feature set to accomplish what they expect of your app.

When the first version of the Facebook app launched with a limited feature set, Facebook users were vocally disappointed, even angry, that essential features were missing like browsing friends' photos, writing on their walls, or commenting on status updates. Later versions added these features, but the app maintained its keep-it-simple focus by dividing the Facebook app into "sub-apps," a concept explored in the Facebook case study on page 236. As always, the lesson is: know your audience and their needs. Give them *just enough* to do what they need to do with your app.



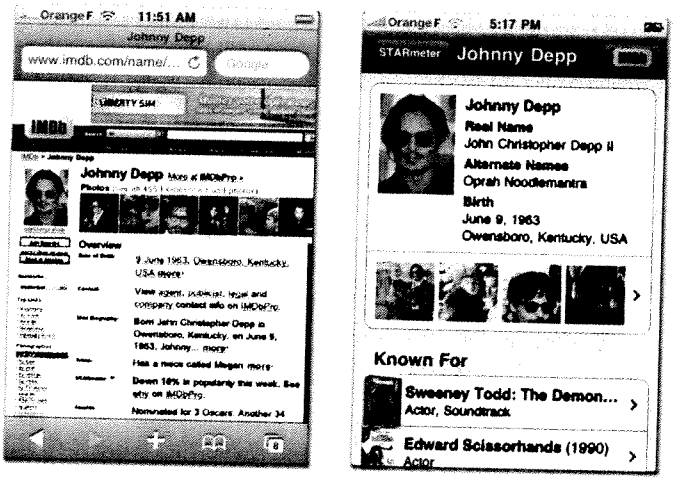
Quickoffice Mobile Office Suite is an example of careful feature pruning. By iPhone standards, it's a complex and ambitious app. Quickoffice lets you read, create, and edit Microsoft Word and Excel files. The desktop versions of Word and Excel are packed with a jillion features and functions, and it would be unwieldy—and probably impossible technically—to pack all of them into an iPhone app. The gang at Quickoffice had to do iPhone triage, supporting only features that would be most frequently used in a mobile context. They made the safe assumption that when you're thumb-tapping the text of a word-processing document, for example, you probably won't futz with formatting complex table layouts, which is work you'd more likely fine-tune later on the desktop. So, while Quickoffice can display tables and images in a document you receive, it doesn't let you edit them—a fair trade-off. Meanwhile, more common formatting options, like bold and italic text, bullet lists, and text alignment are always just a tap away. The result is a useful, compact app that provides all the basics of document editing while gracefully sidestepping more advanced (and unlikely) use cases.

Can't I Get That on the Web?

If you're creating a companion app for your website, chew over how the app might improve on the original site. How is the app better than using Safari, the built-in browser, to view the content online? From your perspective, sure, there are distinct *marketing* advantages to creating a separate app. Repackaging your

website as an app gives you another outlet, a spot in the much-trafficked App Store. And homesteading an icon on a user's phone keeps the content front and center in a way that a Safari bookmark does not. But that's all about *you*, not your audience. If you're a website publisher designing a companion app, think hard about how you can make the app improve on the mobile Web experience, not just duplicate it. Here are a few tips:

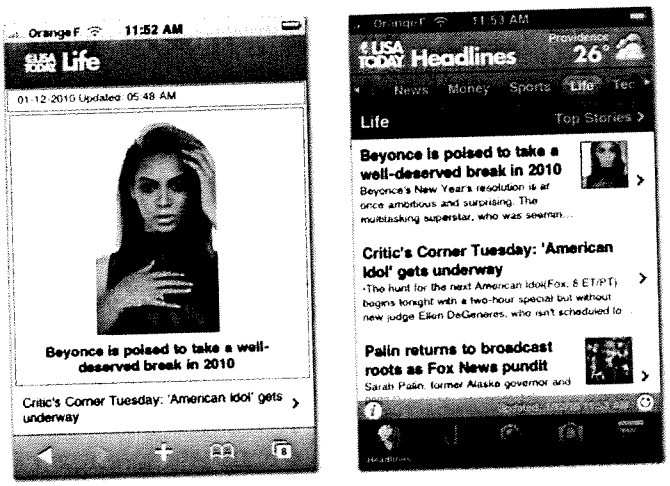
Efficiency is a feature. One of the most important user needs in the mobile context is efficient access. The best website redux apps improve on the interfaces of their Web counterparts. This is especially the case for apps whose companion sites don't yet have a mobile version. Authoritative movie website IMDb, for example, launched an elegant iPhone app for navigating its sprawling film database before ever launching a mobile Web version. The full-sized website at *www.imdb.com* is a hassle to navigate on the small screen, but the IMDb app is a pleasure to use—compact, speedy, and ready to settle pop-culture bets at a moment's notice.



Like most websites, *www.imdb.com* (left) is accessible via Safari, but with a constant dance of pinching and zooming to read each page. Meanwhile, the IMDb app (right) formats the vast site for the small screen, making it a cinch to resolve forehead-slapping "who *is* that?" movie moments.

Native polish makes content shine. A polished interface can set apps apart even from websites especially designed for mobile use. Although the newspaper *USA Today* already had an iPhone-optimized website, for example, the USA Today app provides an even better experience for speeding through news, photos, sports scores, and weather. (See page 90 for a case study of the app's interface development.) While the content for both app and website are essentially identical, the app is custom fit to the iPhone screen with elegant, responsive controls

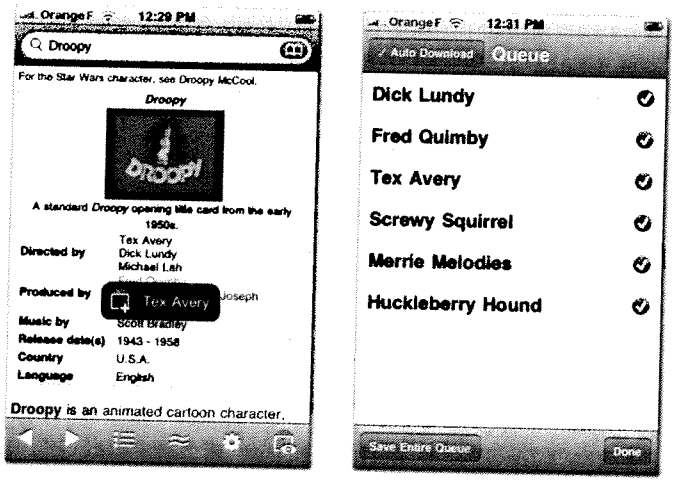
that only a native app can deliver. (Standard web technologies can come mighty close, though. Website owners interested in developing an iPhone app might start by developing an iPhone-targeted website, where most of this book's design principles also apply. *Building iPhone Apps with HTML, CSS, and JavaScript* by Jonathan Stark is a recommended resource on the subject.)



A well-designed app can improve on even a savvy mobile website. The *www.usatoday.com* mobile site (left) is cell phone-sized and built for quick loading over mobile networks. But the USA Today iPhone app (right) provides a much more efficient experience for scanning content and images, packing lots of content into each screen and delivering a strongly branded interface that never distracts.

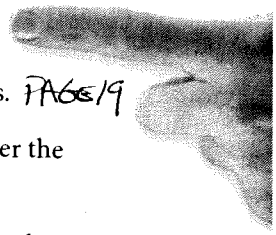
Save it for later. Collecting a stockpile of offline content is an especially tap-worthy feature for mobile content apps. A common difference between a great app and its website counterpart is that the app *caches* content, stashing it on the device for offline browsing. Many news apps, like USA Today, grab all the latest articles when online, saving them so that you can read the news later when you're on a plane, in the subway, or otherwise Internet indisposed. Other apps cache individual pages on demand, adding an interesting new wrinkle to web browsing. Wikipanion Plus, for example, is an app for browsing Wikipedia. Turn on the app's "queue" feature and instead of jumping straight to a new article when you tap a link, Wikipanion adds it to a wanna-read list, the Queue screen. The app downloads these articles in the background, so they're ready when you are—useful over slow network connections or for later, when you have no connection at all. (The feature is also especially well-suited to Wikipedia articles, which are so densely sprinkled with tantalizing links that it requires real discipline to read an article to the end without leaping away. The queue feature lets you line up links to follow later.)

Wikipanion Plus improves on regular web browsing with its clever caching and queuing features. When you tap and hold a link (left), Wikipanion saves the page for later reading instead of opening it immediately. You can review any of these saved pages from the app's Queue screen (right), where downloaded pages are displayed for offline reading.



The upshot: well-considered features and presentation, carefully crafted for mobile scenarios, can turn even content that's freely available online into a tapworthy app for your chosen audience.

Touchpoints

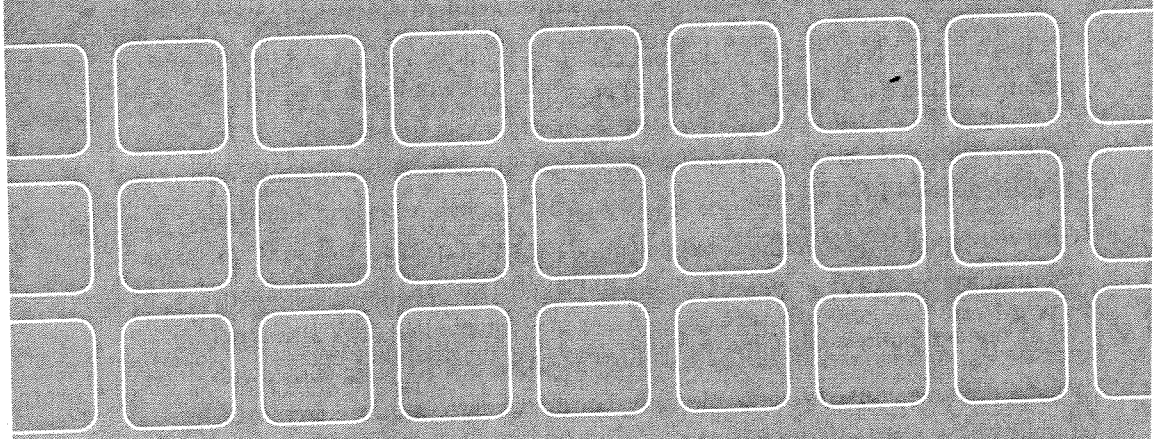


- ✓ Focus on "the five W's" to uncover your app's tapworthy conditions. PAGE 19
- ✓ Focus your app's features for away-from-the-desk contexts. Consider the mobile mindsets: microtasking, local, and bored.
- ✓ Identify what makes your app different from similar apps. "Skills" and "charisma" set apps apart.
- ✓ Give your app a long shelf life by replenishing content or building community.
- ✓ Think big, but build small. Give users *just enough* to do what they need.
- ✓ When building a website companion app, consider how to improve on a mobile website, not just duplicate it.

3

Tiny Touchscreen

DESIGNING FOR SIZE AND TOUCH



AT FIRST BLUSH, the iPhone's tiny size might seem its biggest design challenge. When you're accustomed to creating websites or desktop software for monitors at least a foot across, 3.5 inches is mighty stingy. Yet while designing with a limited supply of pixels is a demanding part of the job, leaving it at that would suggest that your role is only to create efficient screens. The challenge turns out to be more subtle than that, because the iPhone isn't just small: it's handheld and works by touch. That means you're doing something more sophisticated than organizing pixels.

You're designing a physical interface that will be explored by human hands, directly manipulated in a way that desktop software never is. Of course, there's nothing *literally* physical about your app. Your interface is just a virtual representation drawn by so many flickering liquid crystals. But the way it's *used* is physical. Unlike phones or desktop computers, the iPhone has practically no physical interface of its own—no keyboard or mouse, just the single Home button. The device is a blank slate, a palm-sized slab of glass onto which you, the app designer, impose whatever interface you might dream up. You define the physical experience of the device. Your app's buttons may be virtual, but they nevertheless require direct touch and define the device in a very physical way.

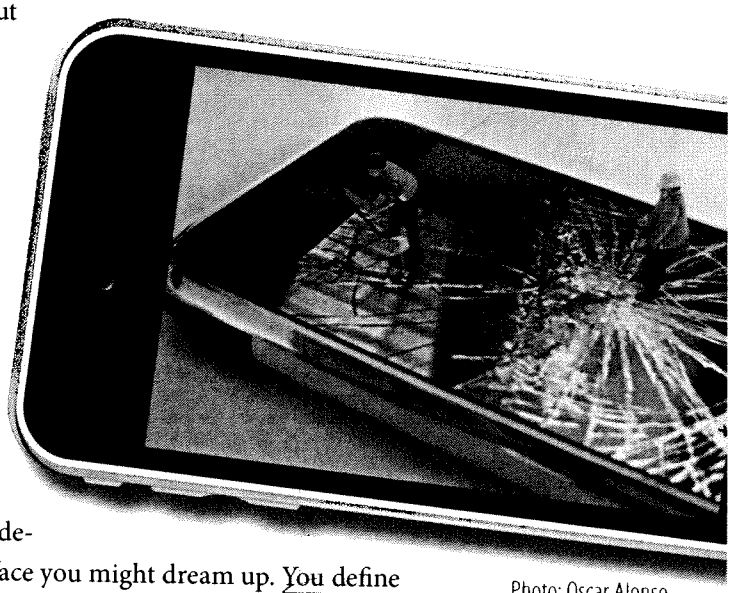


Photo: Oscar Alonso

This means your project is not only a challenge of visual and graphic design but of industrial design, too. Just as surely as if you were soldering circuit boards, molding plastic, or shuffling die-cast buttons, you're designing an app to be manhandled—or at least hand-handled. There are honest-to-god ergonomic issues to account for. Considerations of size and touch combine in iPhone design to present new challenges for interaction designers.

You've already seen how manner and mind-set shape the broad environment for your app and the features you should pursue. The rest of this book deals with a much more immediate set of constraints: the form and conventions of the device itself. This chapter kicks off the process by reviewing the big picture of designing for a tiny touchscreen. You'll explore how fingers and thumbs roam the screen, discover some ergonomic guidelines for comfortable tapping, and wrap up with some good practices for tap-friendly screen layouts.

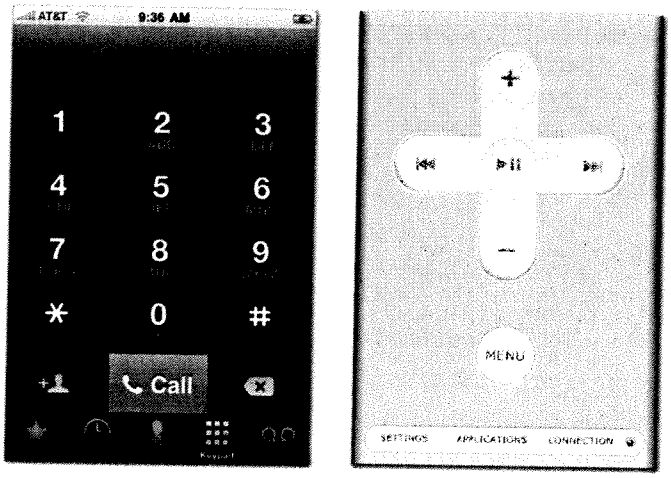
A Physical Feel

More than simply how the app looks, you have to consider how it *feels*. How well does your interface work when used one-handed? Are the most common tap targets within easy thumb range—and what about lefties? Are buttons chunky enough for easy tapping, or does it take surgical precision to hit them?

The tactile nature of iPhone apps is reinforced by the powerful illusion of real-world physics conjured by the iPhone operating system. The iPhone's interactive metaphors are all about tapping, sliding, and flicking—direct manipulation of onscreen objects that respond with lifelike realism. Flick a scrolling list and watch it slow as "friction" takes hold. Fling it even harder against the top or bottom of the screen and the thing actually bounces. Tug the screen down and watch it snap back into place with rubber-band realism. Everything on the iPhone responds to the familiar rules that apply to the everyday physical world: inertia, momentum, friction, elasticity, follow through. The result is an irresistible impression that you're working with real-world objects.

Don't break that spell. More than just tending to appearances—*wow, it looks so real!*—creating an interface that "feels" like an actual device matches expectations created by the iPhone operating system as well as the concrete reality of tapping

away at the hardware itself. (Chapter 8 has more on phone physics and visual feedback, starting on page 257.) Some apps even go all the way with the physical-device metaphor by mimicking the look of familiar gadgets. There are undeniable usability benefits to cribbing a real-world interface that's been used for decades or even centuries, especially when your app performs a similar role to that of the original gizmo. Few will scratch their heads to figure out how to make a call with the built-in Phone app. Its push-button interface conveys its meaning immediately (even three-year olds, for better or worse, know how to dial a phone), and the keypad layout also has the benefit of proven ergonomic success in a handheld device.



Interfaces that mimic physical devices benefit from intuitive familiarity and proven ergonomics. Here, Phone and Rowmote, respectively, crib the telephone keypad and Apple's physical remote control.

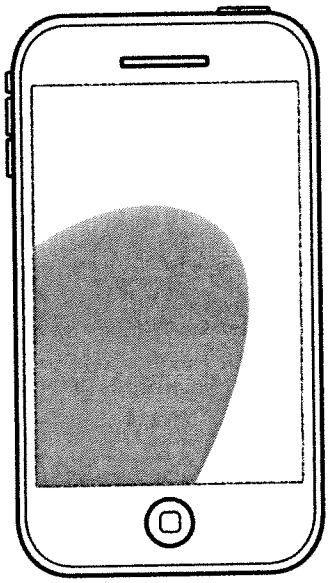
You'll find more discussion of interface metaphors in Chapter 6, but whether or not your app actually apes the look of a mechanical device, that's exactly what it is in practice. Your app's screen design is the sole interface for the iPhone, a gizmo that works by hand, and you have to address the demands of a physical device. When fingers do the walking, designers have to clear the way for them.

Organizing the layout of an iPhone interface means organizing for fingers. More precisely, it means organizing for *the thumb*, since that's the digit that gets the workout when you work the iPhone one-handed. The iPhone is sized perfectly for use in a single hand, allowing your thumb to sweep easily from one corner to the other with only a modest stretch. And because you *can* use the iPhone with one hand, most of us very often do. Whether you're hoisting a coffee, hauling a baby,

or eating your lunch with that other hand, you can still keep tapping away with one thumb to answer email, make calls, or browse the web. Your design should optimize for this one-thumbed tapping.

Rule of Thumb

Thumbs are marvelous. It's our thumbs, along with our affection for celebrity gossip, that separate us from the beasts, but they do have limited range and flexibility. While a thumb can manage to sweep the entire screen, only about a third of the screen is in truly effortless territory—at the side of the screen *opposite* the thumb. For a comfortable ergonomic experience, you should place primary tap targets in this thumb-thumping hot zone. (We'll focus for now on right-handed users, but hang in there, lefties, we'll get to you in a sec.)

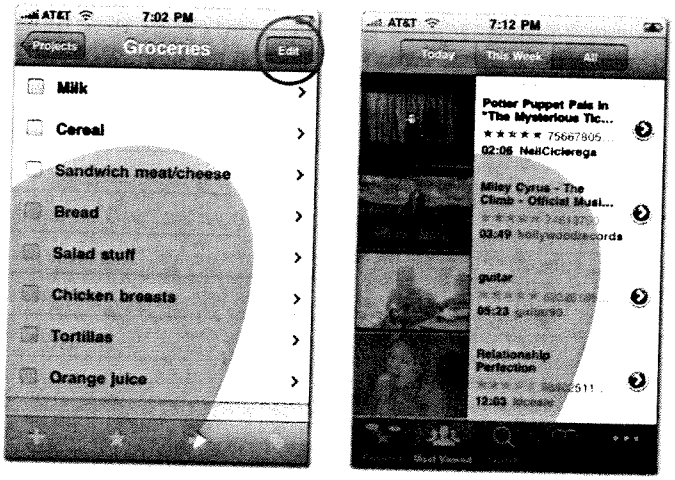


The comfort zone for the right thumb falls on the opposite side of the screen, at the left edge and bottom of the screen. (The top right and bottom right corners are the toughest thumb zones for right-handed users.)

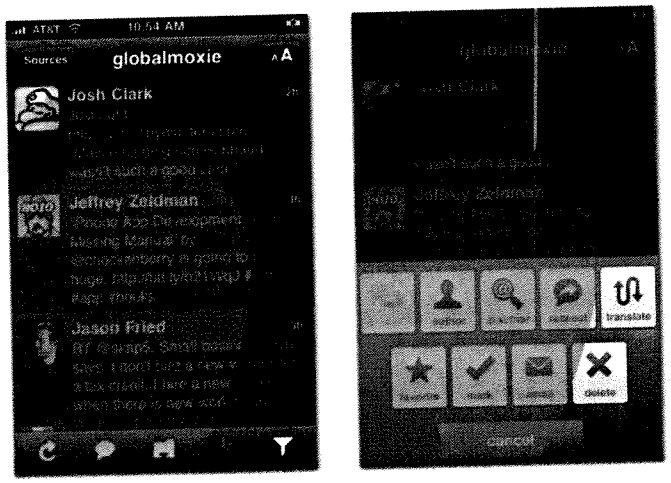
That's an important reason why toolbars and tab bars always go at the bottom edge of the iPhone screen—the opposite of what we're accustomed to for traditional screen interfaces. Desktop software conventions put menus at the top of the screen or window, and websites typically position primary navigation at the

top of pages. Our limited thumbspan, however, flips that convention on its head. Navigation and primary tap targets sink to the bottom on the iPhone. This tap zone gives you hints about how to organize the visual hierarchy of tap targets. Frequently used buttons and navigation tabs should occupy the bottom left of the screen, while lesser used buttons and those that make changes to data can be tucked safely away at top right. The Edit button for changing, deleting, or reordering list items, for example, is conventionally placed at top right, putting it in easy view but also in an isolated and relatively difficult spot to tap, making accidental changes less likely.

The standard iPhone toolbar (left) and tab bar (right) always go at the bottom edge of the iPhone screen in convenient thumb-tapping range. To-do list app Things (left) puts the Edit button at top right safely out of accidental tap range. At right, YouTube's left-edge video thumbnails are chunky tap targets in the hot zone of right-handed users.



Let your thumb point the way in laying out your screens according to the most common use cases for your app. Twitterrific, an app for Twitter users, organizes its buttons according to this thumb-thinking hierarchy. The two left toolbar icons respectively refresh and post tweets, reflecting the two most common Twitter-related activities, reading and posting. When you want to do something with an individual tweet, tapping the toolbar's asterisk icon summons a set of buttons whose thumb convenience likewise reflect their importance—commonly tapped buttons at left, less common at right. "We think a lot about where things get placed based upon ergonomics," explains Craig Hockenberry, Twitterrific's lead developer. "The Delete button is off to the right, the hardest location to tap for right-handed users, and we put other options where your thumb has to work less in order to get to more commonly used actions." (For more about Twitterrific's design, see the developer close-up on page 205.)

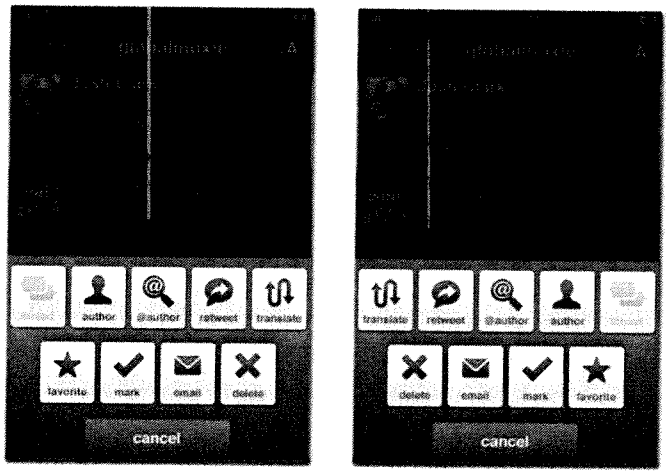


Twitterrific orders its toolbar icons so that the most frequently tapped buttons appear in the right thumb's hot zone, highlighted here in red.

Happily, this organization happens to coincide with the way we read—our eyes naturally scan onscreen menus from left to right, just like the written word. So, for right-handed use at least, physical and visual considerations align. But what about lefties? The hot zone for the left thumb is, naturally, the mirror image of that of the right, which means left-to-right organization makes for awkward tapping for southpaws. When you optimize for right-handed people—about 85 to 90 percent of users—you're actively inconveniencing left-handed folks (as well as righties who switch to the left hand while they scribble a note).

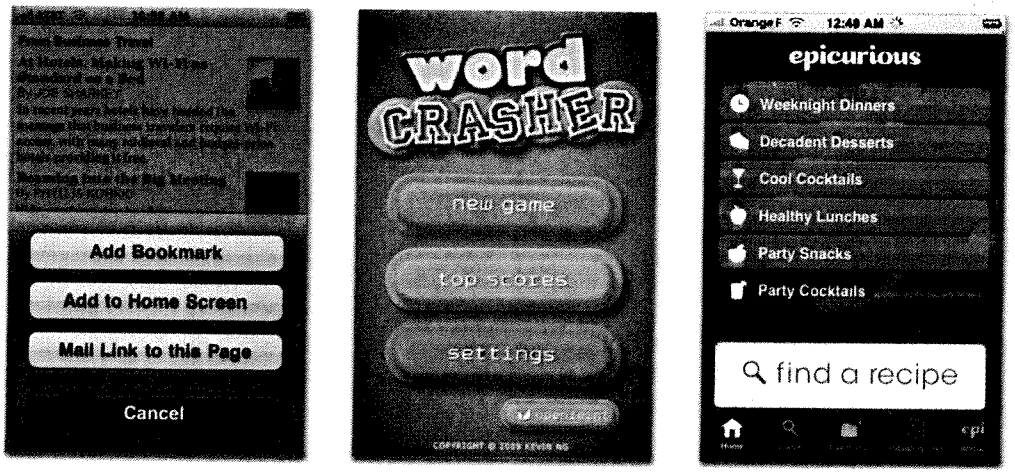
Some conscientious apps, including Twitterrific, go so far as to include a setting for lefties to flip the layout, putting common controls back in the thumb's hot zone. For very tap-intensive apps, this may be a worthwhile strategy. Consider a calculator, for example, which is all about constant tapping. Like Twitterrific, the scientific calculator app PCalc includes an option to flip the keypad layout to accommodate lefties (see page 69). There are downsides to this approach, though. Ordering the most important tap targets from right to left reverses the way we visually process information, asking lefties to burn a little extra brain power to take in your interface. (This effort is repaid with improved ergonomics for frequent taps once they grok the layout, of course.) Perhaps more important, a left-handed option adds an additional preference to your app's settings, which as you'll see starting on page 176, should be pared to a minimum. Finally, there's the addition of a modest amount of code to do the visual switcheroo on your app's controls. Whether it's worth the extra work to ease the thumbstrain of 10 percent of your

Twitterrific offers a setting for left-handed controls, flipping the normal button layout (left) to a mirror layout (right) better suited to lefties.



users depends, as always, on the app. A complex, button-heavy, tap-intensive interface might suggest more mercy for lefties than apps that require less manual manipulation.

In interface design as in politics, sometimes it's better just to meet in the middle. Many standard iPhone controls, including buttons and list items, span the entire width of the screen, an equal-opportunity layout for both left and right thumbs. When space allows, full-width controls are the way to go—an important reason, for example, that wide buttons are cooked into the layout of *action sheets*, the



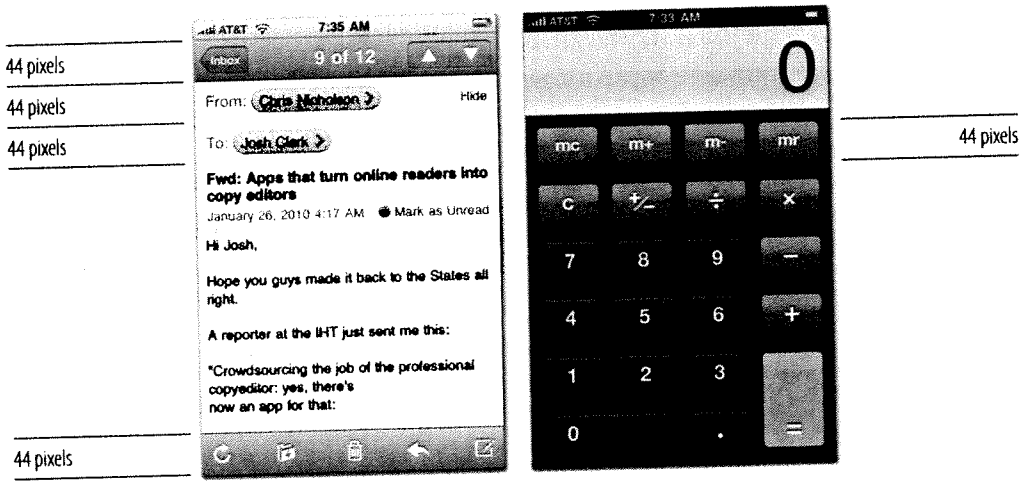
Full-width buttons, like the ones in the standard action sheet (left) or the WordCrasher and Epicurious apps, make for easy tap targets no matter what hand you're using.

iPhone's standard multiple-choice button views (page 167). Big chunky buttons not only give clear guidance to users, they also provide can't-miss tap targets no matter what hand you're using.

Size always matters in interface design. Big text, small text, giant buttons, or tiny ones—they all provide visual cues to what's important on the screen, gently guiding the eye to the next appropriate action. On touchscreen devices, though, fixing the right size for each tappable element is even more important, since every button and control has to be fitted to the finger. Make the buttons too small and you create an exercise in tap-and-miss frustration. The more important or frequent the action, the larger the associated target should be. Big buttons win.

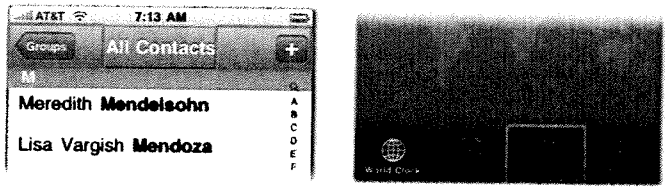
The Magic Number Is 44

But just how big is big enough when it comes to iPhone tap targets? Well, what's the size of a fingertip? Apple pegs it precisely at 44 pixels and this measure appears reliably throughout the standard iPhone controls. In portrait orientation, 44 pixels is the height of buttons in the Calculator app, of the keys of the iPhone's virtual keyboard, of items in a standard list display, of the screen-topping navigation bar, and the list goes on. (With the iPhone's 163 ppi screen resolution, 44 pixels is about 7mm, or just a hair over ¼ inch.)



The 44-pixel block is, in many ways, the basic unit of measurement for the iPhone interface, establishing the visual rhythm of many iPhone apps. That metric is significant as the recommended minimum size to make a tap target (like a button or list item) easily and reliably tappable. Put another way, it's the spread of a fingertip pressed to the screen. By building its standard controls in proportion to that measure, Apple created a device that's not only built *for* the hand, but *of* the hand—measured out in finger-sized units. (In fact, it's wise to craft your overall design to a 44-pixel rhythm, a topic you'll explore on page 75.)

Ideally, any button or other tap target should be at least 44 × 44 pixels. That doesn't necessarily mean that what you see—the outline of the button itself—has to spread over that entire area. Buttons inside the standard navigation bar, for example, are only 29 pixels high, but their *tap area* extends to the full 44-pixel height of the navigation bar. Even if you tap just above or below the button, it still catches the tap as long as you're still inside the navigation bar. Likewise, taps immediately to the left or right are treated as taps on the button itself. Even though the button is *visually* smaller, its tappable footprint honors the 44-pixel minimum, making the button effectively larger than its outline suggests.



Buttons in screen-topping navigation bars (left) have tap areas much larger than their visual footprint. The active tap areas are outlined in red here and span the entire height of the navigation bar, reaching horizontally to the title text. Similarly, icons in screen-bottom tab bars (right) have tap areas that extend several pixels above the tab bar's visible outline.

Apple helps you get this right by providing a whole stable of standard controls that stick to this standard height (you'll explore these built-in views and controls in the next two chapters). When you use Apple's prefab navigation bar, toolbar, or keyboard in your app, its controls automatically use these finger-friendly dimensions. It's only when you start working with custom button sizes and other homebrew controls that you'll need to start counting pixels to make sure you hit the 44-pixel threshold.

As in most things, compromise is sometimes necessary. Even the iPhone's standard controls fudge the 44-pixel rule from time to time. In the keyboard, for example, keys are 44 pixels tall but only 30 pixels wide—similarly, in landscape view, the buttons are 44 pixels wide but 38 pixels high. Apple doesn't have much choice here; it's crucial to include the full QWERTY keyboard in this view, but all the keys just won't fit as 44 × 44 buttons. Something's gotta give. When limited space puts the squeeze on tap targets, here's the rule I've found works best: as long as a tap target is at least 44 pixels high or wide, you can squeeze the other dimension as small as 30 pixels if you really must (these are the same dimensions as a keyboard key). That means, *the practical minimum size for any tap target is 44 × 30.*

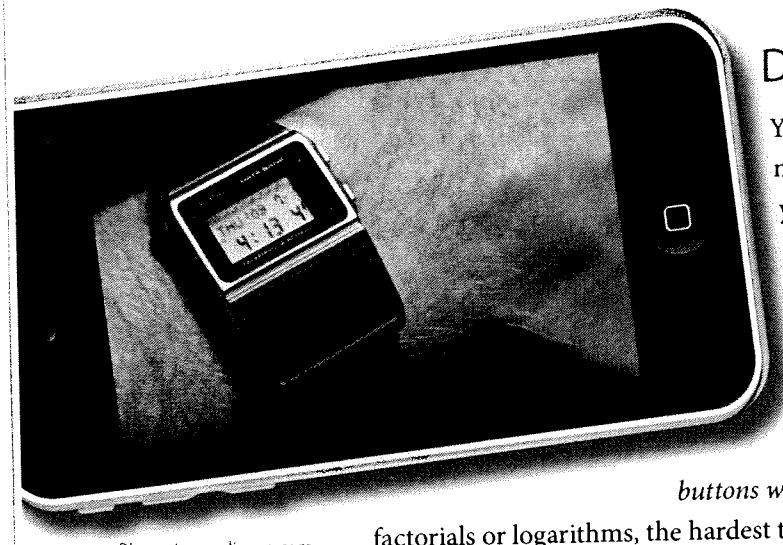


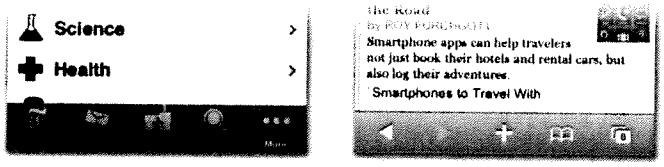
Photo: jonrawlinson.com

Don't Crowd Me

Your faithful author spent many years of his misspent youth with a svelte Casio calculator watch strapped to his wrist before finally retiring it in 1985. The problem wasn't just its tiny controls or its dampening effect on my prom king prospects. *The buttons were too close together.* Forget factorials or logarithms, the hardest thing about doing math on this thing was just hitting the right button. You'd aim for a five, but come up with a two or an eight, who knows—it was more wheel of fortune than calculator. Button size, in other words, isn't the only determining factor of tap accuracy; you have to consider spacing, too.

The iPhone's 320 × 480 pixels make for cozy quarters, and you'll inevitably be tempted to deal with that challenge by crowding the interface. "I'll just nudge these a little closer. I'll just add one more button to this toolbar." To quote a popular phrase of the calculator-watch heyday: "Just say no."

The iPhone's standard tab bar ensures comfortable spacing by limiting app designers to five tabs—no crowding allowed. If you add more than five, the tab bar displays only four of them, adding the More icon which takes you to a screen with additional options, as shown at left in the New York Times app. Apple's guidelines suggest limiting toolbars to just five icons, as shown at right in Safari.



Once again, Apple's standard controls help you do the right thing. For example, when your app uses a standard tab bar at screen bottom to switch between modes or views, the operating system automatically spaces them out for you and limits you to a maximum of five tabs. Toolbars don't impose the same automatic limit, but Apple nevertheless recommends that you limit toolbar icons to five, too. (You'll learn all about tab bars and toolbars on page 106 and page 143 respectively.) In both cases, the screen's 320-pixel width is technically large enough to cram up to seven 44-pixel icons, but they'd bump right up next to each other, introducing the calculator-watch problem for your audience.

It's especially important to give fingers some breathing room at the bottom of the screen. Usability testing reveals this to be a clumsy area prone to mistaps when targets are placed too close to an app's tab-bar navigation. The frustration is

Call Global App and Skype both include phone keypads with buttons that press right up against the tab bar. Call Global (left) makes things especially difficult by making the adjacent buttons narrow and tough to hit, with frequent missed taps. When you want to see your balance, a mistaken tap sweeps you away to the app's World Call screen. In Skype (right), the problem is less pronounced because the big buttons are tough to miss.

compounded by the fact that accidentally tapping a tab bar icon often takes you to an entirely new screen. If you must place targets near the tab bar or toolbar, make sure they're large enough to hit easily. (See how USA Today worked around this problem on page 94.)

Pointed Design

So far, this chapter has emphasized the physical more than the visual, pushing the ergonomic angle of app design. As always, though, form follows function, and all this talk of tap targets and thumb-friendly layouts influences more than just button sizes. When you plan the placement of your app's touch controls based on comfort and ease of use, other aspects of your design naturally fall into place. The best iPhone apps also share a common visual hierarchy broadly following these tap-friendly guidelines:

- Place important info at the top and sink controls to the bottom.
- Design to a 44-pixel rhythm.
- Where appropriate, create at-a-glance displays that avoid scrolling.
- Whittle onscreen elements to the bare minimum.
- Push advanced tools and controls to the background.

Like any set of guidelines, these aren't ironclad laws, just useful practices that can help you quickly block out your design in an appealing, efficient, and accessible layout. The next few sections give you the gist.

Take It From the Top

The most important or frequently used info should float to the top of the screen above the app's primary buttons and controls. This meets our expectations not only of graphic design—headlines at the top—but also the way we hoist and handle just about any physical device. The screen bottom is the most comfortable thumb zone for a handheld gadget, but that's also where the screen is most likely to be obscured by hovering hands. To keep info in clear view, position it above your app's controls. This is a familiar, common-sense layout that applies to most physical devices—iPods, calculators, cell phones, bathroom scales, you name it. Here again, though, it's the opposite of what we'd expect from Web and desktop software where toolbars and menus stake out the top of document windows with primary content below.

Tapworthy apps follow an interface convention familiar with all kinds of mechanical devices. Even when they don't directly mimic real-world gadgets, they organize controls at the bottom and content on top. From left, Cleartune, Weightbot, and iPod.

For primary app navigation, Apple helps you do the right thing by cautioning designers to place tab bars and toolbars at screen bottom (see page 106 and page 143). Likewise, when you summon a keyboard by tapping a standard text field, the built-in keyboard automatically slides in at screen bottom, nudging the typed results to the top of the screen. Follow Apple's lead here: everywhere it makes sense, let the primary controls sink to the bottom and main content float to the top.

This guideline is easy to follow when content and controls are separate, but not all iPhone apps make such a ready distinction. In many apps, the primary display info doubles as a view's main controls. That's especially true for list content;

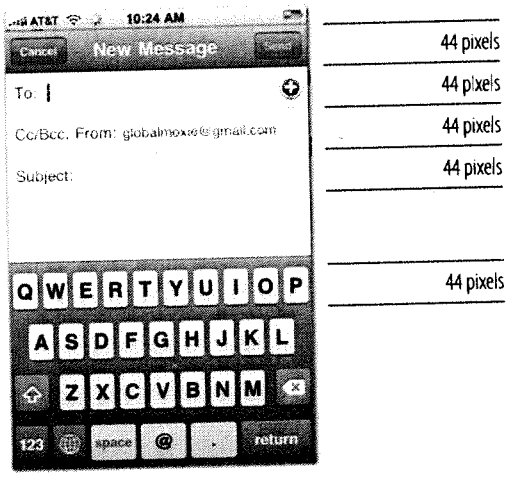
In Gowalla's Passport view (left) every display element doubles as a touch control, but the Check In and New Spot buttons anchor the screen as the most frequent tap targets. Similarly, while the stats at the top of RunKeeper Pro (right) are tappable, the main Stop and Pause buttons remain at bottom, leaving the stats display prominently visible at all times.

whether you're scrolling through contacts, to-dos, songs, or email, you tap list items to see their detail views. In those cases—as well as examples like Home-screen icons or photo album thumbnails—the entire display turns content into tap target from top to bottom. This double duty encourages you to directly tap anything you want to know about, a natural interaction and a good design. Even in these cases, though, it's a good practice to place *the most frequent* tap targets at the bottom, as discussed earlier.

Design to a 44-Pixel Rhythm

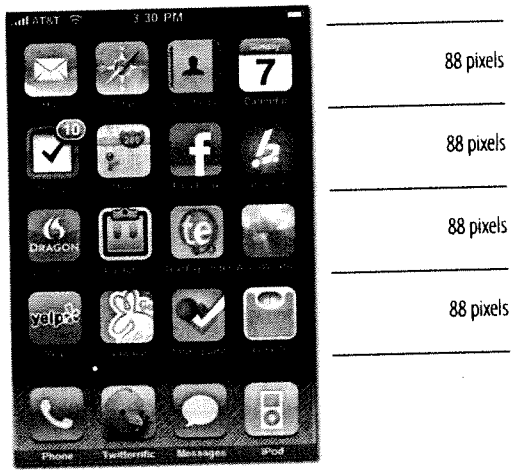
The iPhone looks good, sure, but it *feels* good, too. The industrial design gurus at Apple carefully designed the iPhone hardware to feel right in your hand, with weight and proportions that are as appealing as the device's glossy sheen. The same should go for your app design, with screens that are not only pretty but are constructed in proportion to the hand, with a sense of balance in each element's visual weight.

Part of that, as you've seen, is a matter of using finger-sized tap targets placed in easy reach. But the size of a fingertip can do more than just determine button sizes: as a unit of measure lifted from the hand itself, fingerspan provides a visual building block for giving your interface design a consistent rhythm in natural proportion to your paw and its pointer. By loosely blocking out your design in a



44-pixel grid, you ensure that interface elements are sized in harmony with one another as well as the fingers that work the device. As you saw earlier, several standard iPhone elements—navigation bars, toolbars, list items, keyboard keys—are sized to 44 pixels, establishing the vertical rhythm of many screens.

Go with that flow, at least loosely. You don't have to rigidly stick to aligning every single element to a 44-pixel grid. In fact, you *can't*, since the iPhone's 320 × 480 dimensions don't round neatly to 44, and some of the built-in controls like the tab bar at screen bottom veer into slightly different sizes. Instead, the point is that the



The iPhone Home screen builds on the 44-pixel visual rhythm by organizing icons into 88-pixel rows.

44-pixel measure provides a natural size to build with, one that rhymes with the size of several standard controls. Creating interface elements based on that proportion gives your designs visual balance and stability.

Be a Scroll Skeptic

Another way to reinforce an impression of sturdiness and weight is to build screens that stay put—no scrolling required. When it's possible to comfortably fit a screen's content into a single no-scroll view, you should go for it. The "out of sight, out of mind" effect is especially strong on the iPhone, where distracted mobile users speed through apps. When info isn't front and center, chances are good they won't see it at all. It's a matter of both brain and strain: scrolling requires extra thought as well as extra swipes, and one of your jobs as app designer is to reduce both.

This might seem subtle, but just asking users to *figure out* that they need to scroll requires them to fire up brain cells: "Wait, what's missing, and how do I get to it?" By contrast, taking in a screen's complete content at a single glance lets users focus exclusively on the content without making their gray matter do any background processing about what's offscreen. This might seem like coddling—seriously, it'll break their brains to scroll?—but the best apps bend over backward to reduce the overhead required to work the app itself. No-scroll screens require less brainwork while also reinforcing the illusion of your app as a physical, not virtual, device. A fixed screen gives a sense of solidity.

The point here is not to avoid scrolling at all costs (this section is titled "Be a Scroll *Skeptic*"—not a zealot, reactionary, or dogmatist). Eliminating the scroll is just one tactic for designing a display that's easy to absorb, not a goal in itself, but when it suits your app's content, it's a design tactic that encourages healthy restraint.

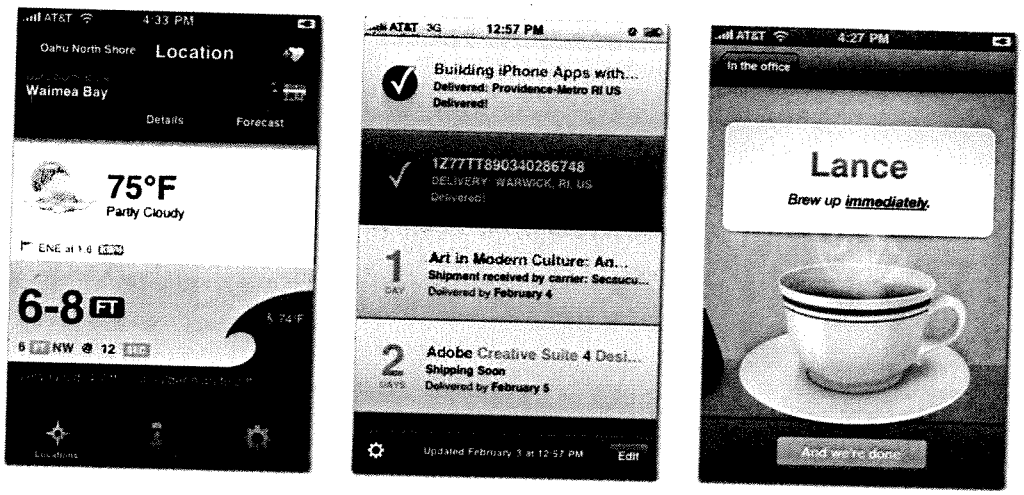
There's a whole category of apps devoted to the single-screen display. Apple dubs them *utility apps*, narrowly focused tools that provide quick summaries or perform a simple task, almost always in simple no-scroll screens. The built-in Weather app is the quintessential utility app. It provides streamlined forecasts for up to 20 cities, each tidily presented in a get-it-quick, no-scroll view—you flip

The Weather app offers a compact weather forecast, summarizing a week's weather in a single screen. The border around the info creates a solid container to signal that there's no additional scrolling content below. The page indicator dots at screen bottom indicate that more screens await (see page 103 for more info about page indicators).

through them like cards. The app's presentation lets you absorb the week's forecast for each city with just a glimpse. The carefully contained layout makes it instantly clear that there's no additional info below the fold, no "Is there more?" thinking required.

The Weather app pulls off its single-screen layout by stripping content down to the bare minimum. A few icons and high/low temperatures hardly tell the whole story of your local weather, a topic that occupies entire websites and 24-hour weather channels. Instead of indulging in complex detail, the app focuses on giving the quick gist in a simple and efficient display. This just-the-basics approach is a good strategy for utility apps, the simplest class of iPhone software. Like Weather, the best of these apps rely on graphically rich displays to telegraph simple info quickly, with big can't-miss text and images that sink in with just a quick peek. Contrary to what you might expect, the success of compact interfaces often relies on big text and chunky images cushioned with generous surrounding space. Apps pass *the glance test* when you can hold them at arm's length and still soak up their info effortlessly.

The glance test reinforces an essential principle of tapworthy app design: *clarity trumps density.* A crowded screen creates more work and confusion for your audience. This doesn't mean you're obliged to chuck your app's detailed info in order to have a beautiful and intuitive interface. It's not an either-or bargain. Complexity itself isn't bad; the trick is making complexity seem uncomplicated.

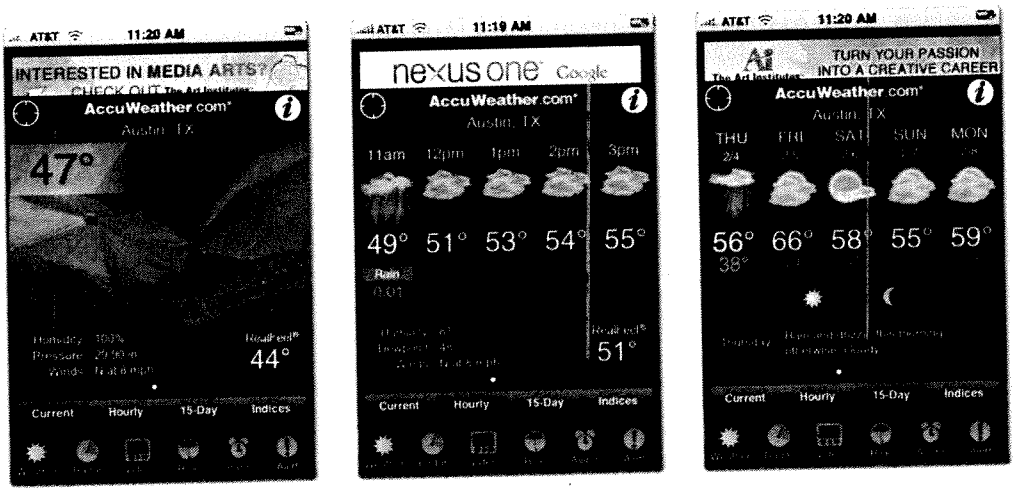


From left, apps like Surf Report, Delivery Status Touch, and Tea Round pass the glance test with high-impact text and graphics that are easy to read even from several feet away. Surf Report shows water conditions, Delivery Status Touch shows package whereabouts, and Tea Round shows whose turn it is to brew the tea.

Even within the iPhone's tiny screen, it's possible for apps to reveal complex information in a simple display—and yes, without scrolling.

The AccuWeather.com app, for example, is a sophisticated alternative to the Weather app that steps beyond the utility category by providing several screens of detailed weather info. Each of these screens (see page 80) is self-contained in a no-scroll display, providing dense information layered in multiple but uncomplicated views. Want to find out what the humidity will be in a few hours? Tap the desired hour in the app's 15-hour forecast and the details pop up immediately. The main screen shows the big-picture overview, and the app tucks additional details behind tabs or icons instead of piling it all into the same screen at once. It's at once intuitive and information-rich, no scrolling or crowded layout required.

Healthy scroll skepticism means recognizing that you don't have to reveal all your information in one shot. It's a bit like the art of conversation: don't be that guy who drones on and on without pausing to check if he's saying something that actually interests his suffering listeners. Especially when there's lots to be said or explored, the best conversations are interactive, with listeners allowed to ask questions instead of passively receiving info. In an app, a tap is effectively a question in the conversation. The best apps provide top-level, need-to-know information



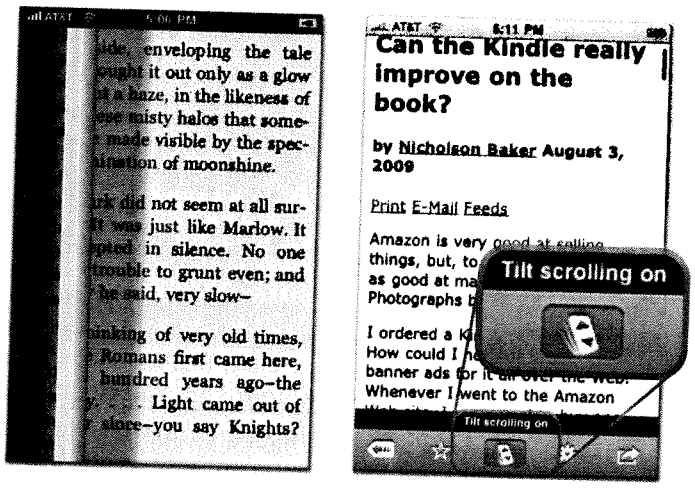
AccuWeather.com's simple tab control lets you flip between current, hourly, and daily weather forecasts. On the hourly and daily views, you can tap the day or hour to get more info—an intuitive way to make data available without crowding too much into the display at once.

at a glance and, from there, let users tap something to "ask" for more information about it. Chances are, you don't need to know what the local wind speed will be at 3 p.m. every time you launch AccuWeather.com, but when you need it, the answer is just a tap away. In the meantime, the main view still gives you the basic at-a-glance info that you'd get from the built-in Weather app.

Are all these extra taps really better than scrolling? Just on the basis of physical comfort, a tap is an easier gesture than the swiping scroll. A more important consideration than tap quantity is tap *quality*. As long as there's an appropriate reward waiting on the other side of every tap, extra taps are okay—and usually less of an imposition than a long scrolling screen. If you focus the main screen on the most important tools and info, you can safely tuck secondary content into another view. If done right, tapping through fixed screens requires you to do less visual scanning than it takes to locate content on a scrolling page. Content and controls on a fixed screen remain reliably in the same place visit after visit, so it's easy to duck in and out of an app to get the info you want without pausing to get your bearings.

Even long-form content allows creative alternatives to scrolling. Most apps for reading books, for example, use a page-turning metaphor instead of scrolling to

advance the text. This lets you tap the screen just once to flip the virtual page to a fresh screen of prose, sparing you constant swiping while also reinforcing the illusion of handling a physical object—convenient and familiar. Instapaper Pro, an app for saving and reading lengthy online articles, likewise offers an option to page through screens a tap at a time. The app also offers scrolling but with a clever ergonomic gimmick to spare you swipe-swipe-swipe tedium: you can scroll just by tilting the phone back and forth to advance the text.



Stanza and Instapaper Pro offer thumb-sparing alternatives to swiping through long text. Like most ebook readers, Stanza (left) lets you tap the screen to flip to a new page of text. Instapaper (right) offers a tilt-scrolling feature to advance text by physically tipping the device.

But let's not be strident anti-scrollers. While it's good to avoid scrolling where appropriate, it's not like it's inherently evil. It's part of the fun of the iPhone's physics, and it's obviously essential to some apps. To-do lists, news feeds, articles, and emails inevitably run long, and scrolling is (usually) the best way to handle those kinds of long-form content. List-based interfaces that try to wriggle out of a scrolling screen often just feel awkward (see the interfaces the USA Today designers experimented with on page 91).

When your app does require scrolling, just be sure to keep the primary app controls anchored in one place. In early versions of the Facebook app, for example, the tab control for flipping through a friend's profile content was itself part of the content area, which meant it scrolled out of view with the rest of the page. When you finished reading through a pal's status updates, for example, you had to scroll back up to the top of the screen to switch over to your friend's photos—assuming that you could even remember where you last saw those tab controls. Facebook 3.0

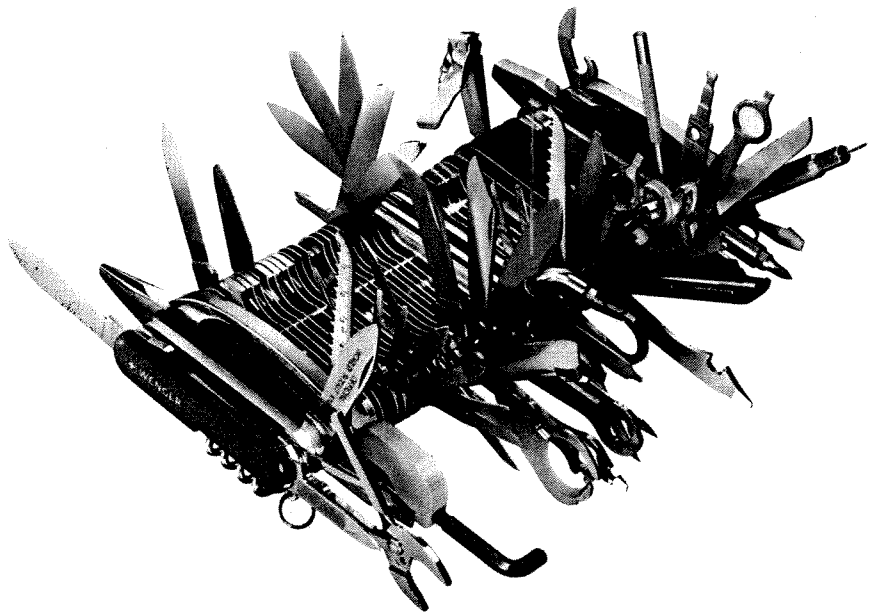
remedied this where'd-it-go problem by anchoring the wayward tabs to the bottom of the screen, saving both time and head scratching. The lesson: scrolling or no, a view's primary controls should never skitter offscreen. Anchored elements create a sense of stability and consistency.

In Facebook 2.0 (left), the control for switching between wall posts, profile info, and photos was part of the scrolling content area and could disappear offscreen. In Facebook 3.0 (right), the tab control is always anchored to screen bottom.

Ergonomic and visual simplicity should be important goals for your app design. Whether or not you ultimately decide to include scrolling screens, approaching the scroll with skepticism asks you to be more discerning about what you include. Committing to the iPhone's 320 x 480 footprint puts limitations on your design as firm as if you were designing a real-world device. When there's only so much room for your tools, controls, and content, you have to ask yourself a useful question: "Do I really need all this stuff?"

Edit, ~~Edit, Edit~~

The Guinness world record for "most multifunctional" pocket knife belongs to Wenger, the company behind the storied Swiss Army knife. The company says the knife's 87 gadgets (including a laser pointer, cigar cutter, and golf reamer) can be used for no fewer than 141 functions. The $1400 gadget is a nifty feat of Swiss ingenuity and—who knew!—Swiss humor, too. Alas, weighed down by its three pounds of gizmos, this "most multifunctional" knife has no practical function at all, a pocket knife that doesn't fit in your pocket. This slice-n-dice Goliath



was, of course, never really designed to be used. It was a novelty created for the company's 100th anniversary, a whimsical project to bring together every gadget the company ever included in its knives. At a certain point, as Wenger's craftsmen added more and more tools, the knife suddenly stopped being a knife, and it just became a doorstop.

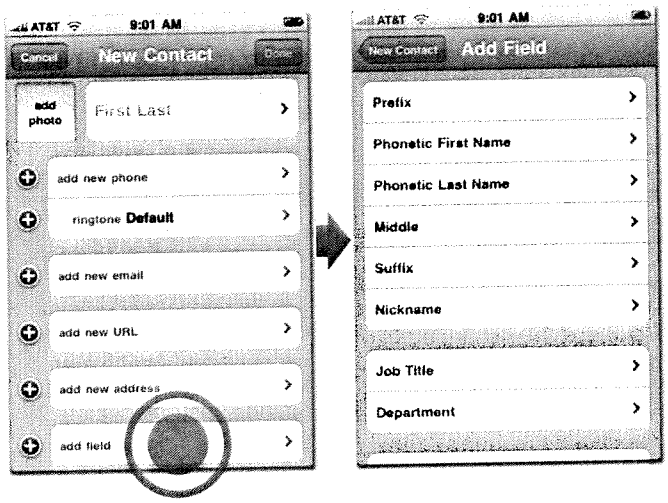
In geekspeak, you might call this a crowded interface. While the knife is obviously (and intentionally) ridiculous, it's a winking reminder that somewhere in the reptile part of our brains, a misguided instinct tells us that more is always better. More features, more preferences, more flexibility—the gizmo that does the *most* is always the best. In the end, of course, the best gizmo is the thing that lets us do what we need to do with the greatest ease. In mobile devices—whether it's a Swiss Army knife or an iPhone—that almost always means removing features and *chrome*, the buttons, icons, and other controls that often crowd an interface.

The last chapter encouraged you to trim your app's functions in the planning stage, whittling the app to a sharp focus on the most important features. Apply the same tapworthy filter to every onscreen element, too. Include controls only if they're likely to be used most of the time by most of your audience. Be ruthless when you consider every button and icon: Does this element invite attention? Is

it clear what it does? Does it deliver something meaningful? Every tool should be tightly related to the primary task at hand, and auxiliary tools and content should be dispatched to a secondary screen or sliced out altogether. Text labels should be terse and words shouldn't be repeated.

It might seem harmless to add just one more icon, but every onscreen element comes with a cognitive cost for your users. It takes longer to scan the screen, longer to absorb the possible options, longer to figure out what you're supposed to do. Don't make 'em think. When every pixel is precious, your app doesn't have room for question marks that will snag users as they hustle through. It's like the pocket knife: it's easy to choose a blade to use when there's only one of them, but it takes forever when there are 87.

Just as you saw in the AccuWeather.com app, you don't have to show every last scrap of information all at once. Consider whether second-tier content and controls might be dumped entirely and, if not, nudge them over to a separate view in your app. The built-in Contacts app does this, for example, when you add or edit a contact. The main screen is a no-scroll view of the most common contact info; the app doesn't pester you about less frequent fields and instead provides an "add field" button at the bottom of the screen. From there, you can choose from among 12 additional fields to round out your contact info when the standard fields don't do the trick.



When you add a new contact, the Contacts app displays only a few primary fields (left). Tap the "add fields" button at screen bottom to go to a secondary screen (right), which offers 12 less common fields.

JOBS' ELEGANT SIMPLICITY

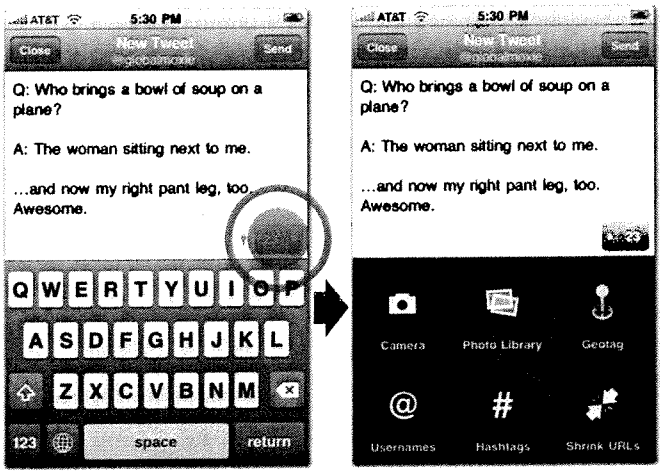
Apple is particularly clever about doing more with less. The company's software has a well-deserved reputation for both elegance and ease of use. It looks great, works intuitively, and drips with good taste. Like all good editors, Apple's designers do this in part by removing distractions and plucking out features that aren't relevant to mainstream users in order to focus more screen real estate and polish on the features that matter to the most people. What's not included in Apple's built-in iPhone apps is as important as what is.

Secret Panels and Hidden Doors

Here's the thing: leaving stuff out inevitably drives some people nuts. Power users will, in particular, pine for all the tools and flexibility that mainstream users don't miss. For many, the basic Weather and Stocks apps just don't cut it. If your app addresses an audience with more advanced needs, then you'll almost certainly have to add more features, tools, and content than your app's mainstream counterpart. That doesn't give you a pass to create mind-numbingly crowded interfaces. The same rules apply, and the pocket knife still has to fit in the pocket. If your app really does have to include those extra features and controls, you've still got to prevent the extra interface chrome from elbowing aside the main content.

This is a challenge that the many iPhone Twitter apps wrestle with. On the surface, Twitter couldn't be simpler: post terse updates and read those posted by others. Yet an enormously complex infrastructure of tools, social customs, and third-party services has emerged around this basic concept. Your tweets can link to photos or videos, point to your physical location, mention other Twitter users, "retweet" someone else's post, embed #hashtag metadata, and more. The dilemma for Twitter apps is how to make a dead-simple interface for firing off a quick tweet while also surfacing the many power features that make Twitter megausers tingle. The brute-force approach is to pile a bunch of icons and controls into the tweet-posting screen. Trouble is, when you add a keyboard to the mix, all that extra chrome quickly squeezes out the most important thing: the text you're typing. It's a sign of trouble when controls leave little room for the content.

The Twitter app (formerly known as Tweetie) solves this problem by hiding all those add-on controls behind the keyboard. The button that displays the

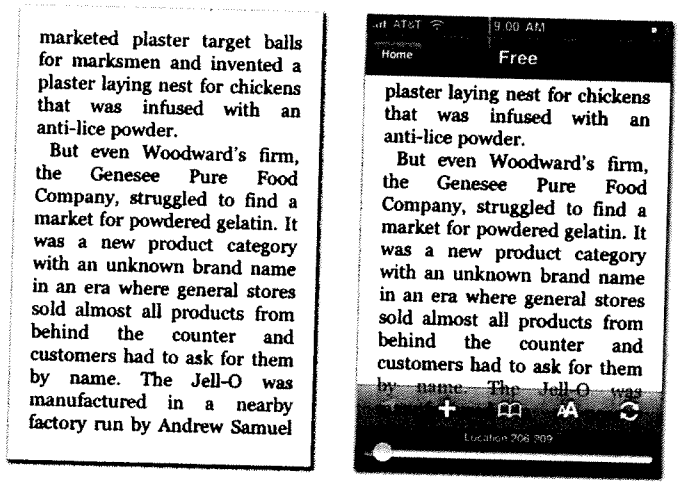


In the Twitter app, tapping the character-count button (left) slides away the keyboard to reveal a control panel (right), a compact means to put advanced features within reach without crowding the screen.

remaining character count for your tweet doubles as a toggle switch that slides the keyboard out of the way to reveal all the options you can apply to your tweet—a secret control panel for power users. When you're done, the keyboard slides back into view, and you're back to typing, no extra chrome in sight.

Other apps use similar tricks to move interface elements out of the way. Ebook apps Stanza, Kindle, and Eucalyptus all hide their main controls completely to allow book text to fill the screen when you're reading. Ditto for the Photos app when flipping through full-screen pictures. Hiding primary controls like this admittedly brings a risk of disorienting people, leaving them uncertain about how to escape the current view. In all of these examples, though, tapping the screen summons the controls again, a tactic that even panicked users will try soon enough. It's a legit approach, but one that should be used sparingly and only by apps that get a big benefit from an unsullied full-screen view.

Advertisements also count as interface chrome, and unlike primary controls, most users wouldn't mind seeing them slide away for good. Ads present a business and usability dilemma: they take up valuable real estate and present content to which many users are actively hostile, yet many free apps rely on them to fund the whole operation. The sliding panel approach provides a useful compromise. In the USA Today app, for example, a 50-pixel ad banner slides up from the bottom of the screen when you first arrive at an article screen. It holds its place for a

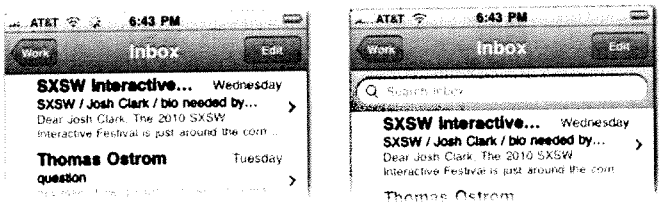


Kindle banishes its controls to turn over the full screen (left) to your book. Tapping the screen brings the controls back (right).

few seconds before sliding out of view—a reasonable compromise for both advertiser and reader.

The point here is that for all of this chapter's talk about designing your app to feel like a real-world device, it's okay to bend the rules to take advantage of your app's virtual environment. Virtual keyboards and sliding panels let your app do things no physical gadget could do by swapping out the interface for a new set of tools or content. These "cheats" let you layer secondary tools and content into your app without crowding the main screen. Turns out it's not all that different from the Swiss Army knife after all, a handy gadget that can constantly become something new. While pesky considerations like size and weight prevent the Swiss Army knife from offering too many tools, the iPhone's virtual interface gives you a theoretically unlimited interface surface area. Just keep flipping through secret panels as you need them, right?

Well, yes and no. The trouble with hidden content is, you guessed it—it's hidden. You'll continue to bump against the "out of sight, out of mind" problem. The more secret panels you spirit away, the more overhead you ask people to take on to remember how to find them. You have to take care to provide visual cues to help them find their way. If you decide to use a hidden door, that means you have to put the latch in clear sight to make it easy to open. Ideally, this trigger should be well-labeled or at least in such close proximity to the content you're working

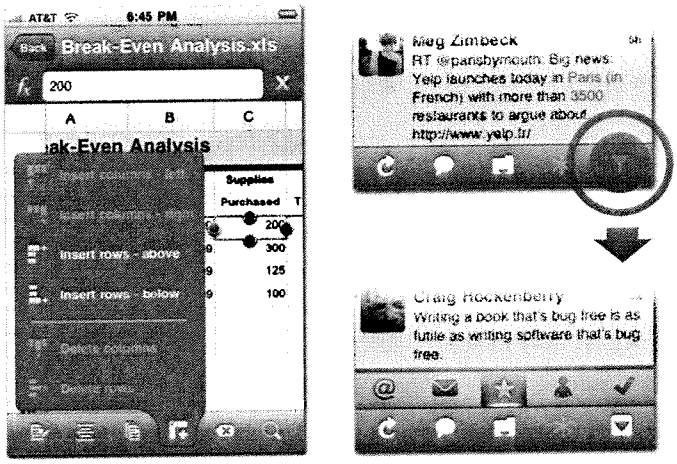


The search bar is hidden offscreen in the Mail app until you tug the screen down to reveal it.

with that it suggests what you'll find on the other side. The latch for your hidden door should look tapworthy and invite action.

A more discoverable approach is to hide interface chrome in plain sight. Most apps that have a search feature position their search bars at the top of the screen, premium real estate. The built-in Mail app, among many others, reclaims this valuable space by scrolling the view so that when you first land on a screen, the search bar is tucked away just out of view. If you want to search, you tug the screen down or tap the screen-topping status bar to zip the search bar into view.

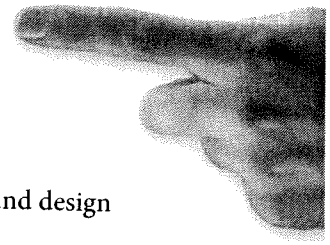
Another strategy for discoverable secondary controls is to offer toolbars that emerge temporarily from the app's main navigation. Quickoffice's iPad-style pop-over menu provides tools for formatting text, for example, and Twitterrific's filter bar lets you choose the type of tweets you'd like to view in your Twitter timeline. As always, the goal is to keep your interface visually uncomplicated, no matter how complex your app may be. Limit interface chrome, but hide only as much as you have to. Give all of your features and controls a hard look before you include them in your design to make sure they're really tapworthy. If they pass muster, be



Quickoffice (left) offers secondary tools in popover menus when you tap an icon in the main toolbar. Twitterrific (right) similarly reveals icons in a second toolbar when you tap the funnel-shaped Filter icon at far right.

sure to place each element carefully in intuitive, easy-to-find locations, hidden or not.

Touchpoints



- ✓ Ergonomics matter: consider how your app feels in the hand.
- ✓ Put primary controls in the thumb's "hot zone."
- ✓ Forty-four is the magic number. Make tap areas at least 44 pixels, and design to a 44-pixel rhythm.
- ✓ Be generous with space and don't crowd your design.
- ✓ Feature primary content at the top, controls at the bottom.
- ✓ Keep the main controls within easy reach, and avoid scrolling where practical.
- ✓ Reduce interface chrome by dispatching power tools to secondary views with secret panels and hidden doors.