

## Designing Graceful, Gracious Interfaces for iPad

- 起 — *iPad is the biggest shift in the way people use computers, ever*
  - We have a rapidly growing audience.
  - Our new audience is trained by iPhone and the web to expect straightforward, foolproof interactions.
  - We can't get away with providing designs that put the burden on the user to make things work.
  - At the same time, remember that people are smart and adaptable; they can pick up *reasonable and useful* new concepts without blinking.
  - It is a lot like the rebooting of the way people perceive technology that was the original Macintosh.
  - A set of constraints that knocks some sense into us and makes us think again about what our software is really for.
- 承 — *iPad requires a combination of new and old user experience thinking*
  - **We are going to be held to a whole new standard** of agreeableness and usability.
  - There are at least **two wrong ways to take advantage of the iPad's bigger screen**.
    - 1: Take your iPhone app and **blow it up** to fill the space. Woo! iPad app done!
      - We briefly considered doing this for OmniFocus while we worked on a real iPad app. If we had released something like that, sure, it would have been a slightly more pleasant experience than using the iPhone app blown up to 2x. It also would have sent a clear message that we do not understand the iPad.
    - 2: Take your iPhone app and **add junk** to fill the space. Woaaa! So many more features!
      - The design process for OmniFocus for iPad included so many moments where it would have been easy to just throw another button in the toolbar and let people figure it out.
      - For the item editor, it took serious thought and argument to make it clear why we couldn't just make all of the fields and controls visible at once, given all that generous screen real estate.
      - But the beneficiary of the iPad's expanded screen real estate shouldn't be *me*, the designer. It belongs to the user, to display their data and their content and their work as generously as possible. We should leave as much of that space to them as we can.
      - To spray controls and chrome all over the place puts a cognitive burden on the user every time they try to do something: which of these twenty controls is the thing they need right now?
  - **What we did instead.** [break this down]
    - Instead, we took the UX reconsideration opportunity to offer *fewer* controls at once in our item editor. We divided the controls by meaning — organization, dates, notes, attachments — so that

people could choose the sort of control they're looking for from a short list, then choose from a short list of controls (or a single big field, in the case of the note). Everything is presented in understandable chunks, progressively disclosed.

- Why did we divide the controls by meaning rather than by commonality of use? Because any claim of commonality was too presumptuous to expect everyone to agree with. Some people set a context on every item; others eschew contexts entirely.
- It's tempting to count taps. We get concerned about how many taps it takes to do a certain task. This is almost always misguided. **Taps are cheap.** When iPhone was introduced, its approach to interaction seemed so tedious: go to another screen, go to another screen, open a menu of buttons, tap here, tap there, tap tap tap tap tap tap. But when you're actually in it, you realize that it's not such a burden. Taps are so much cheaper, cognitively, than the abstraction of shoving a hunk of plastic around on a tabletop to make a little arrow maybe approach a thingy on a screen, then make sure you've positioned it properly, then push down a plastic button. On iOS, it's see thing, hit thing. It's not a hassle; it's kinda fun. Removing that layer makes such a difference: the burden of each poke is small enough to let you get away with requiring twice as many pokes as on the desktop to do the same job.
- Another reason you can't just blow up your iPhone interface, or just extrapolate its complexity out to the edges of your new, bigger screen: now, parts of the screen aren't very close to the user's hands. If they're holding the iPad in one hand, or if it's on a desk or their lap, it's pretty easy to poke stuff anywhere on the screen with their index finger. But don't count on it. There's a good chance they are holding the iPad in the more comfortable two-hand position. Then, their thumbs can pretty easily reach a zone around the edges of the screen, and the center takes some position-shifting to get to. It's not onerous, so the effect is probably mostly subconscious. But if you can put most of your important stuff in those steering wheel zones, your app will feel that much more pleasant.
- **Forgiveness** has always been an important principle; now it is crucial.
  - You won't get the benefit of the doubt when something doesn't behave as expected.
  - Software is less and less perceived as an inscrutable and capricious magic that occasionally demands sacrifice.
- **Brutally massacre features** until only the very fittest are left standing.
- **Immediate feedback** is one of the most important, but easiest, ways to make your app *feel good*.
  - Most people don't consciously think of it, but they definitely feel the tiny bits of uncertainty that pile up every time they have to wait 250 milliseconds to find out if something worked.
  - If a button doesn't depress or highlight or glow or anything when you tap it, your app feels slow, unresponsive, and unpredictable; it feels like it belongs on some *other* touch-screen platform.
- It might seem rude to obscure content with popovers, contextual menus, sheets, and other such stuff. But as long as that stuff gets out of the way at a moment's notice, with a single, nearly-unconscious tap-away, **it's fine to obscure content**.
  - Those things are transitory — they exist for precisely as long as the user is using them, and then they go away. So don't worry about it too much.

- Hover was nice to have on the desktop. You could see where people were thinking about interacting, and reassure them: yeah, that thingy is clickable, and if you click it, here's the sort of thing that might happen. **We don't have hover anymore.**
  - We've found a few things that work similarly though.
  - First, though, tappable stuff needs to look pretty tappable. Button borders, little shadows, contrast with the surface around it.
  - In our canvas-based apps, where there are several different things you might want when you tap on an object, we proceed step-by-step through them with each tap: select the group, show the contextual menu, select the member of the group, show the contextual menu, and so on. This sounds tedious, but remember that taps are cheap, and people are very good at tapping again when they don't immediately see what they want.
  - If an interaction requires a hold or a drag, we have a little response that emerges on touch-down and then recedes on touch-up. This almost always encourages people to try touching and holding, or touching and dragging.
- Multi-touch is awesome. You can get all sorts of intuitive, tactile, fun gestures to do all sorts of stuff, so cool, woوو. Not really: we learned pretty quickly that **we shouldn't require multiple fingers or rapid successions of taps for anything important.**
  - Lots of times, the user is holding their iPad in a way that would make anything but the simplest gestures a pain.
  - You could really only count on pinch and rotate, Apple's main existing multi-touch gestures, to be discoverable and usable, and even those can be unreliable.
  - Leave the multi-touch stuff as equivalent to keyboard shortcuts: power-user ways of getting at stuff that's more conventionally offered in buttons and menus.
- **Where to find wisdom**
  - Sure, read the HIG.
  - An anecdote about a WWDC UI consulting session.
  - The HIG is like The Elements of Style: a big list of pretty decent suggestions, that seem like strict rules and make you constantly worry you might be breaking one, when you should be focusing on making something delightful.
- **An anecdote about the plain drag**
  - We spent so much time on what an ordinary drag ought to do in our canvas based apps. Do you want to draw a line? Drag out a regular shape? Draw the outline of a shape? Move an existing object? Pan the view? After lots and lots of arguing, we arrived at a workable system. There's a drawing mode, with its own very distinctive visual style and an obvious way to exit it, where drags create objects. And there's the normal manipulation mode: one tap selects an object, another tap selects the next object behind it, a double-tap edits its text. A *touch*, *hold-for-the-slightest-instant*, *drag* moves an object; an immediate swipe, or any drag on a blank area of the canvas, pans the view. No resizing objects with pinching, no three-finger-double-tap to select the object behind, or any of that.

- 転 — *iPad will change the way we make other software too*

- On the desktop we already strive to **avoid dumping preferences on the user**. iOS has taught us that we should strive even harder.
  - What feels like an affordance for different ways of working is often an unwillingness to make a decision, and the foisting of that decision onto tens of thousands of people around the world.
  - Generally, I have found that almost all defensible preference settings are regarding one of two things. Keyboard behavior, because the keyboard is such a personal interactive abstraction. And technical details that arise out of different people having different hardware setups.
  - For everything else, it's better to find an unambiguous, sensible, learnable way for your product to work. This is really really hard, and really really worth it.
- **Offer sensible defaults and templates.**
  - Work super hard on making sure that if the user does no customization whatsoever, they end up with something pleasant.
  - It's not enough to offer the tools necessary to make it beautiful, though you should provide those too. If most people never even need to know those tools exist, then your software is that much easier and you win.
  - This is another case where a medium-small amount of effort by a few people can save a medium-large amount of effort times *so many people*.
- **The interface should disappear as much as possible.**
  - The desktop makes it so, so difficult to concentrate. Layers and layers of windows, bars on the edges of the screen full of barely-recognizable status information, alerts demanding that you attend to technical matters.
  - You don't feel like the computer exists to help you get things done — you feel like it's barely tolerating your feeble presence.
  - iOS apps are quiet and respectful. If you say you want to draw a picture, you get a canvas and maybe a couple of drawing tools. Lots of times, you can even make the tools disappear, leaving you with *nothing but your thoughts and your work*.
- **Your user's content and work is more important than any part of your UI design.**
  - Get your chrome out of the way as much as you can, so that the content can be the star.
  - It shouldn't have been, but this was kind of a shock to the way we design software. *Of course* the chrome should be unobtrusive! *Of course* it should disappear!
- **Be skeptical of purely quantitative judgements**
  - It seems responsible to be scientific about how we build things. Usability testing is important. But don't favor observations about task duration or frequency of taps override your instincts and experience, or forget to consider how the interaction *feels*.
  - A design choice that makes an action take an extra tap, or requires an extra moment's glance, but makes the software make more sense and feel more right, is a good one.

- 結 — *More than ever, iPad requires us to design with respect*
  - Every time we cut down stress, frustration, or wasted time, we're saving lives
  - iPad leaves you alone with your work.
  - The touch interface puts you closer to your work than you've been since we started using computers.
  - The product of this whole process isn't software; it's work, play, learning, and people's lives.