APPLE INTERFACE MYSTERIES

by MICHAEL E. COHEN

$9.99

Click here to buy the full 173-page “Apple Interface Mysteries” for only $9.99!
# Table of Contents

**Read Me First** ................................................................................................................. 4  
  Updates and More ............................................................................................................. 4  
  What’s New in This Version ............................................................................................. 5  

**Introduction** .................................................................................................................... 6  

**A Brief Introduction to Usability** ..................................................................................... 9  
  What Usability Is .............................................................................................................. 9  
  What Affordances Are .................................................................................................... 11  

**Mac Puzzles Old and New** ............................................................................................... 14  

**How Mac Menus Got Complicated** .................................................................................. 16  
  The Case of the Overburdened Memory .......................................................................... 16  
  The Riddle of the Menu Shortcuts .................................................................................. 18  
  The Mystery of the Mercurial Menus ............................................................................ 21  
  The Puzzle of the Popping Menus ................................................................................ 23  
  The Perplexity of the Popovers ..................................................................................... 27  

**How Mac Keyboards Got Complicated** .......................................................................... 30  
  The Modifier Key Adventure ............................................................................................ 31  
  The Alternative Character Enigma .................................................................................. 36  
  The Input Source Imbroglio ............................................................................................ 40  
  The Substitute Text Surprise ........................................................................................... 42  

**How the Mac Mouse Got Complicated** ........................................................................... 46  
  The Which Mouse Affair ................................................................................................ 46  
  The Reverse Scroll Incident ............................................................................................ 53  
  The Matter of the Multiple Fingers ................................................................................ 56  

**How the Mac Desktop Got Complicated** ......................................................................... 61  
  The Task Trouble ........................................................................................................... 61  
  The App Pursuit ............................................................................................................. 66  
  The Space Case ................................................................................................................ 76  
  The Edge Case ............................................................................................................... 82  

Click here to buy the full 173-page “Apple Interface Mysteries” for only $9.99!
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Ineffable Evanescence of iOS</td>
<td>87</td>
</tr>
<tr>
<td>The Case of the New Kid in Town</td>
<td>88</td>
</tr>
<tr>
<td>The iPhone and Its Early Limitations</td>
<td>88</td>
</tr>
<tr>
<td>Along Came iOS (and the iPad)</td>
<td>90</td>
</tr>
<tr>
<td>The iOS and iPadOS Look and Feel Casebook</td>
<td>93</td>
</tr>
<tr>
<td>The Affordance That Hid in Plain Sight</td>
<td>93</td>
</tr>
<tr>
<td>The Hidden Button Escapade</td>
<td>99</td>
</tr>
<tr>
<td>The iOS and iPadOS Edge Cases Casebook</td>
<td>108</td>
</tr>
<tr>
<td>The Home Button That Vanished</td>
<td>108</td>
</tr>
<tr>
<td>The Great Swipe Hunt</td>
<td>109</td>
</tr>
<tr>
<td>The iOS and iPadOS Touchy Text Casebook</td>
<td>121</td>
</tr>
<tr>
<td>The Dancing Fingers Oscillation</td>
<td>122</td>
</tr>
<tr>
<td>The Glass Keyboard Discombobulation</td>
<td>129</td>
</tr>
<tr>
<td>Crime Fighting Tips</td>
<td>146</td>
</tr>
<tr>
<td>Takeaway Retakes</td>
<td>146</td>
</tr>
<tr>
<td>Mystery Solving Resources</td>
<td>161</td>
</tr>
<tr>
<td>About This Book</td>
<td>170</td>
</tr>
<tr>
<td>Ebook Extras</td>
<td>170</td>
</tr>
<tr>
<td>About the Author</td>
<td>171</td>
</tr>
<tr>
<td>About the Publisher</td>
<td>172</td>
</tr>
</tbody>
</table>

Copyright and Fine Print ................................................................173
Welcome to *Apple Interface Mysteries*, version 1.0.1, published in April 2020 by alt concepts inc. This book was written by Michael E. Cohen and edited by Joe Kissell.

Having trouble figuring out what to type/tap/swipe/click/drag/say to your Mac/iPhone/iPad/AppleWatch/HomePod? It’s not just you. Legendary for ease of use, Apple technology can sometimes—often—stymie both novice and advanced users as they try to get seemingly simple things done. This book explores some of the not-so-easy-to-figure-out things you sometimes have to do to get your Apple devices to do your bidding and explains how things got so complicated.

If you want to share this ebook with a friend, we ask that you do so as you would with a physical book: “lend” it for a quick look, but ask your friend to buy a copy for careful reading or reference. Discounted classroom and Mac user group copies are available.

Copyright © 2020, Michael E. Cohen. All rights reserved.

---

**Updates and More**

You can access extras related to this ebook on the web (use the link in *Ebook Extras*, near the end; it’s available only to purchasers). On the Ebook Extras page, you can:

- Download any available new version of the ebook for free, or buy any subsequent edition at a discount.

- Download various formats, including PDF, EPUB, and Mobipocket. (Learn about reading on mobile devices on our [Device Advice](#) page.)

- Read any entries we may post to the ebook’s blog.

If you bought this ebook from the Take Control website, it has been added to your account, where you can download it in other formats and access any future updates.

Click here to buy the full 173-page “Apple Interface Mysteries” for only $9.99!
What’s New in This Version

Version 1.0.1 of this book is a minor update to fix a handful of typos and formatting infelicities.
Introduction

“It’s not you, it’s Apple.”

When I first talked with Take Control publisher Joe Kissell about the idea for this book, that sentence came up as what they call in my home town of Hollywood the book’s high concept.

Let me expand on that a little. For decades, Apple has been legendary in the technology industry for its ability to make the complex simple, to package incredibly sophisticated technology into easy-to-use packages. And as with many legends, this one contains an element of truth, even though the legend itself is more legend than true. Apple’s devices may, on the whole, be easier to use than those of competitors, but easier isn’t the same thing as easy when it’s hard to figure out.

A couple of cases in point:

• To find out how what percentage of a battery charge you have left on your shiny new iPhone, you simply drag down from the upper right screen corner. That’s easy. But why drag from just there? How would you figure that out?

• To save the Numbers document that’s open on your Mac with a different file name, just hold down the Option key as you pull down the File menu and choose “Save As.” Easy. But why hold down Option? Because otherwise the Save As command isn’t on File menu —how would you figure that out?

This book explores the chasm that can lie between easy to use and figuring out how to use it when it comes to your Apple devices. For, as we all can testify, how you get things done on an Apple device can oscillate wildly between the blindingly obvious and the nearly impossible to figure out, even when what you finally end up doing to get the thing done is, in itself, easy.

This book is not a litany of the usability crimes that Apple has committed over the years (although that might be a fun book to write). It’s a
book about puzzles (which you can solve on your own with thought and patience) and mysteries (which, in order to solve, you need outside assistance from clues, tips, and luck). Note that I call both of these things mysteries in this book, because an unsolved puzzle looks exactly like a mystery.

In this book I first provide a high-level overview of what usability and user-centered design are. This is background information that you may skip, but, since this is a book about usability, I suggest you skim it at least.

After that I turn to the Mac and look at four standard Mac features that began simple and became complicated: menus, keyboards, mice and other pointing devices, and the desktop.

From the Mac, I turn to iOS and iPadOS. Once again, I start with some history, after which I explore how the touch interface on these devices become richer and more confusing, how increasingly important their screen edges have become, and how their text editing capabilities have evolved.

I end with a recap of the “takeaways” that I have sprinkled throughout the book, offering additional advice and thoughts, followed by a short list of places to go for more information to help you unravel Apple mysteries on your own.

While in this book I do look at cases where Apple’s easy to use comes with a softly muttered once you solve the mystery, the book is not about particular mysteries. What I hope it will do is help you understand why Apple (and other developers—it’s not just Apple) have ended up constructing usability mysteries, and it tries to give you a head start on solving others. It’s a little bit example, a little bit history, a little bit theory, and a little bit rock ‘n’ roll.

I can’t promise at the end of the journey you’ll be able to solve every puzzle you might need to solve or unravel every mystery you need to unravel to bend your Apple devices to your will.

But you should end up knowing how to solve a lot of them.
Amaze your friends!
Impress your family!
Read on...

Click here to buy the full 173-page “Apple Interface Mysteries” for only $9.99!
A Brief Introduction to Usability

Consistency and predictability are admirable qualities in both app interfaces and tech book series. That’s why I see the irony of opening this book with some abstract stuff—a bit of background, some conceptual material—instead of diving right into the how to get things done stuff that you’d ordinarily expect to find in a typical Take Control book.

However, you need the abstract stuff because this book is about understanding how and why your apps and devices act the way they do, and understanding that requires context. Don’t worry: it’s only a very little bit of background and barely a handful of concepts.

Specifically, this short chapter briefly explains What Usability Is, which has a lot to do with determining how you and your devices interact, and even more specifically, it describes What Affordances Are, which end up involved one way or another in nearly all of the mysteries you encounter as you use your devices.

What Usability Is

Usability is the field that studies how people use things, and it attempts to learn and to prescribe more efficient and effective ways to use them. As software and digital devices have become increasingly more important economically, so, too, has the field grown. Today, most major software and hardware developers have one or more teams of usability specialists working with the software and hardware engineers; in some cases, the usability specialists are engineers themselves.

Note: In the early days of the Mac, and even before, Apple had a usability specialist, Bruce Tognazzini, whose Apple Human Interface Guidelines literally guided the way that many Apple II and Mac apps interacted with their users.
Software usability specialists tend to do two sorts of things:

- **Categorize interactivity methods and patterns**: This activity trickles down to developers in a lot of ways. A fair amount of it comes directly from Apple in the form of software frameworks (for example, Apple’s UIKit framework), and as usability guidelines. It can also arrive indirectly—for example, in the form of the free apps that Apple provides. These apps, such as Pages and Keynote, not only give Apple’s customers useful software but also give Apple’s developers actual working examples of various interface methods and patterns of interactivity.

- **Test (and evaluate) interactivity patterns and methods**: It’s not a mystery why usability testing is big business: usable interfaces are profitable! A successful user interface means fewer user support calls and emails that developers have to field, and all those support calls and emails cost money. Usability studies generally focus on evaluating the following (you may want keep this short list in mind as we look at the various usability puzzles covered in this book):
  
  - How easy something is to learn
  - How efficient it is to get something done
  - How easy it is to relearn it after being away
  - How pleasant it is to use

**The Takeaway**: As random and arbitrary as some user interfaces can seem, most user interfaces don’t just happen. Lots of people, working hard and thinking hard, consciously and deliberately craft them. When interfaces fall short of perfection it’s seldom from want of trying. As Osgood Fielding III says at the conclusion of *Some Like It Hot*, “Nobody’s perfect!”
Two things that caught everyone’s eyes when the Mac debuted were its screen—full of pictures instead of indecipherable lines of text—and its mouse. With the latter, an unfamiliar device one pushed around on the desk in order to move a pointer arrow on the screen, a Mac user could manipulate those pictures on the screen. It was a “graphical user interface” and it was all about visual affordances: windows you could drag, buttons you could press, sliders you could slide.

Of course, the Mac that came out of a bag in January 1984 looked a lot different from any of today’s Macs...but not everything about it was completely different. Look at what was on its screen (Figure 1) and then look at what’s on yours today. The windows don’t look that much different, there’s still an Apple symbol in the screen’s upper-left corner, the Finder still has a menu bar with File and Edit menus across the top, and (if you have your Dock visible at the bottom of your screen) there’s still a trashcan icon in the lower-right corner. As much as the Mac interface of bygone days, it’s chock-full of affordances.

![Figure 1](image-url)

**Figure 1:** If you are reading this book on your Mac, compare this early Macintosh screen to your screen. Notice any similarities?

Click here to buy the full 173-page “Apple Interface Mysteries” for only $9.99!
In fact, today’s Mac interface has become a rococo reflection of the old 1984 Mac interface. Those complications and coruscations didn’t appear by whim, although certainly some marketing whimsy was involved—I certainly haven’t forgotten the “lickable” buttons of Mac OS X 10.0 that Steve Jobs famously touted.

It’s simply that, over time, as the Mac became capable of doing more and more, the simple graphical interface of 1984 became as much an obstacle as an advantage. Driven by a combination of customer requests, developer ingenuity, and the work (sometimes inspired, sometimes not) of its own usability team, Apple continually added new interface techniques to help people do more things more efficiently—and more pleasantly—on their Macs.

What’s more, as Apple created new devices, it created new ways of doing things on those devices.

In turn, these popular new devices and their new ways of doing things have led Apple to bring some of those new ways back to the Mac from those devices. These days, after all, many new Mac users come to buy a Mac after having owned and used another Apple device first, and it makes both good marketing and good usability sense to give these new Mac users some familiar ways to get things done.

Three and half decades on, doing things on the Mac has become more complicated.

This next few chapters tour some of the more puzzling places of interest in the complicated landscape that is the modern Mac interface. In them we’ll look at How Mac Menus Got Complicated, How Mac Keyboards Got Complicated, How the Mac Mouse Got Complicated, and, finally, How the Mac Desktop Got Complicated.

Note: Speaking of places of interest, the official Unicode name of the puzzling % symbol that most Mac people pronounce “command” is the “place of interest sign.”
How Mac Menus Got Complicated

The original Mac’s menus and its always-present menu bar were a spiffy thing in 1984. Up until then, most computer users had to memorize lots of commands, and then laboriously type those commands to get something done.

Menus blew away that memory burden. Instead of having to be stashed in your memory, nearly all of an app’s commands were stashed in menus, categorized neatly. Want to know what file-handling commands are available? Pull down the File menu and take a look.

That worked. Until it didn’t.

The Case of the Overburdened Memory

Menus are great—until you have an app with many hundreds of commands tucked away in its menus. At that point, you’re no longer asking yourself “what is the command for...” but “where the heck did I see the command for...”.

As it turns out, Apple’s usability folk came up with a way of answering that question years ago, although it’s not an answer that most people know about or use. Part of the Mac usability guidelines that Apple offers to developers is one that specifies the standard Help menu that apps should have, and that suggests the items the Help menu should have on it. One of those items, if the developers have built their apps using Apple’s frameworks, is an automatic Search field at the top of the Help menu (Figure 2).
Figure 2: The Help menu helps you find commands on other menus; here, Nisus Writer Pro shows me where the Open Recent command hangs out.

And what gets searched? Among other things, the app’s menus: type part of a command’s name in that field and the app shows you the menus where that command might be.

Granted, finding and using this capability isn’t much of a mystery... except that, until you use it, there’s no indication of what using the Help menu’s Search field searches. And, of course, you must remember a command’s name well enough to search for it in the first place.

Welcome back, memory burden!

Note: And speaking of memory burdens, consider hierarchical menus, some of which can go many levels deep. These are both clumsy to access and almost impossible to remember. They’re not a mystery, but they do add to the memory burden an app can impose.
How Mac Keyboards Got Complicated

Of course the very first Mac had a keyboard; what self-respecting personal computer of the 1980s wouldn’t? Today, to our sophisticated 21st century eyes, the keyboard that shipped with the 128K Mac in 1984 looks quaint, but certainly not alien: even though the Mac keyboard was not the primary device for issuing commands to the computer (that’s what the Mac’s menus were for!), it was quite well suited for the other main task for which computer keyboards were designed—data entry.

True, it did differ from other computer keyboards of the day by lacking arrow keys and a Control key and by offering several keys that were new-ish to keyboard users of the era, like the Command (⌘) key.

But it wasn’t that mysterious a device, except for one quality that all computer keyboards have in common:

*A computer keyboard is just a big collection of buttons.*

And that has implications when it comes to discoverability and predictability, two important ingredients in a satisfying user experience. Here’s why:

- On a mechanical typewriter keyboard what a key does is predictable. A key is directly connected to a typebar that has a couple of characters molded into its metal end: you press the key and the typebar flies up and strikes the platen. Pressing the key with the “a” label always raises the typebar that has an “a” molded on it. Easily discovered, easily relearned.

- On a computer keyboard, a key can signal either that it is being pressed or is not. That’s it. What the computer makes of that information is entirely up to software. The symbol inscribed on any
particular key on a keyboard suggests what you may get when you press it, and it usually can be trusted—but not always.

And that’s where the fun begins.

The Modifier Key Adventure

Apple provides a variety of keyboards for Macintosh users today, but even the most minimal of them include the arrow and Control keys missing from the original Mac keyboards, and a fair number more. Aside from the character keys, on any recent Mac keyboard you can find Command, Control, Option, Function (usually labeled “fn”), Caps Lock, and, of course, Shift.

All of these keys (usually) do nothing on their own when pressed. Instead, they change what other keys do in conjunction with them. Such keys are usually called modifier keys and they’ve been around for a while: one of the earliest was the “Meta” key that first appeared on some early computer keyboards (it looms large in nerd legend).

The Shift modifier key has been around the longest. It is a key every typist knows: when held down, it makes a letter key produce an upper-case instead of a lower-case letter. There’s little mystery to what it (usually) does.

Trivia: Why is the Shift key called that? Because on mechanical typewriters holding that key down literally shifted the position of the typebed that held the typebars. That, in turn, changed which of the two characters molded at the end of a typebar made contact with the platen when the key attached to the typebar was pressed.

But what about the other modifier keys? What does Control control? What kinds of options does pressing Option provide? What are the commands that Command commands?
How the Mac Mouse Got Complicated

Point and click. Sounds simple. But over the years, the way Mac users and their mice have interacted has changed, encompassing many additional gestures—even if you don’t count the *digitus impudicus* that frustrated users often employ. And the mice themselves have changed as well, or been outright replaced by other kinds of pointing devices. We’ll look at mice first.

The Which Mouse Affair

For more than two decades, the Apple mouse remained functionally changeless, even as its exterior look went from the original boxy beige design (which, in fact, predated the Mac, having first seen light with the Apple Lisa) through physical changes of height, weight, shape, and color, and its sensor evolved from a primitive physical rubber-ball sensing mechanism to optical ones (first, LEDs, then lasers). Wikipedia has an *informative article* on the variations in the Apple mouse line over the decades, so I’m not going to go over all the entertaining details here, nor reminisce about the many happy hours I spent degriming mouse balls and shopping for the perfect mouse pad. But I am still going to spend some time on Apple mouse history because you can’t always understand where you are until you see where you have been (skip ahead to Trapping the Minimal Mouse if history is not your cup of tea).

One Button to Rule Them All

Regardless of the many mice Apple produced in the Mac’s first two decades, you need to know this salient fact: from the first Mac mouse to leave Cupertino in 1984 until 2005, the one thing all Mac mice had in common was *a single button*. This, in itself, was a departure from
the three-button prototype mouse that the device’s inventor, Douglas Englebart, debuted in 1968 (as Englebart explained, it had three buttons because “[t]hat’s all that we could fit”).

Why only a single button? Many reasons, but two stand out: it was cheaper to make, and it was easier to teach people how to use. And that wisdom held true for 21 years. What changed?

Lots. The biggest two changes were Apple’s move away from the “classic” Mac OS to its UNIX-based OS X, and Apple’s introduction of the Mac that may have saved Apple from bankruptcy, the iMac and its USB interface for connecting peripheral devices:

• The first change brought in a whole new population of Mac users: UNIX users, who were very tied to their multiple button pointing devices and chafed under the Mac’s draconian restriction to a single button.

• The second change brought in a whole industry of pointing device makers who were happy to see the proprietary Apple Desktop Bus device connectors be replaced by a standard that worked for computers other than just Macs. This change allowed these device makers to sell the same devices to Mac users that they sold to Windows users—and those Windows users were used to having mice with more than a single button.

While it is true that even before Apple’s USB transition a number of vendors produced alternative pointing devices, from mice to trackballs to drawing tablets (Take Control’s own publisher used to work for one such vendor), most Mac users continued to wield the single-button mouse that Apple provided. The biggest exception among them were users who toted around Macintosh PowerBooks, which employed a built-in trackball and button. In fact, the early PowerBooks had two buttons to go with the integrated trackball—but both buttons offered the identical functionality.

Once Apple finally eschewed its single-button-mouse convention, it did so in a surprising way: by producing a new mouse (which Apple first dubbed “Mighty Mouse” before trademark problems compelled it to
How the Mac Desktop Got Complicated

As we’ve seen, increasing complexity is the leitmotif in the history of the Mac’s mouse and keyboard interfaces, so there’s no reason to suspect that the Mac desktop interface would not sing the same tune.

The Mac desktop was designed as a graphic metaphor for an actual desktop, one where you could open folders, take out documents to work on, or drag them into the Trash, which you could find sitting right on the desktop as was customary...yes, that’s right, even at the beginning the desktop metaphor was a little strained, and that strain only got worse as the years progressed.

The Task Trouble

The original Mac could run only one app at a time—along with itty-bitty-mini apps known as Desk Accessories, like the Calculator, the Notepad, and the Puzzle desk accessories similar to what you might find on any busy executive’s real desk.

The Birth of Mac Multitasking

The “one app at a time” part was the first big problem with the desktop metaphor. In the real world, people who work at desks tend to work on more than one task at a time, but not so on the original Mac: when you opened a document to work on it, it covered the desktop completely. To see your desktop again, you had to stop working.

These early limits were understandable: the first Macs barely had enough storage and processing power to handle even one app at a time. But new technology improves very quickly, and soon the one-app-at-a-time constraint on the Mac was a design instead of a performance constraint. The Mac user interface and its underlying software were
not designed to run multiple apps at a time, even when the hardware got powerful enough to do so.

Several early schemes attempted to work around that. One of the first attempts was Switcher, which could keep more than one app at a time running. However, each app took over the entire screen: when you switched apps, the app you had been using was replaced on the screen with the next app, which slid in to cover the first. After Switcher, though, came MultiFinder, which finally did keep the desktop visible behind the windows of the currently running apps. But MultiFinder, which continued being a standard part of the Mac’s operating system until Mac OS X came along, multi-tasked...well, badly is the best way to describe it. Users had to fiddle with allocating memory to each app manually (virtual memory was not a Mac thing) and if a single app crashed, they all did (memory protection was not a Mac thing).

**How the Mac Look and Feel Became the Face of Unix**

Mac OS X’s arrival in 2001 rebuilt the Mac user interface on top of Unix. Except, well, it wasn’t *the* Mac user interface, it was *a* Mac user interface. It looked similar to what was now called the “Classic” Mac interface, but it was far from identical (**Figure 31**).
Apple was no stranger to handheld devices by the time Steve Jobs, with the new iPhone in hand, appeared onstage at Macworld Expo in 2007—the company had already released the Newton and the iPod. The Newton, though loaded with promising technology and interface ideas, had proved to be an ambitious failure, but the iPod, which offered simplicity as its most important interface achievement, was so big a success that at its height it brought in almost as much revenue as Apple’s entire Mac line. Even more importantly, the palm-sized music player was putting Apple-branded technology in many more customers’ hands than its flagship Macintosh ever had. By the middle of the 21st century’s first decade Apple could clearly see that small handheld devices would be a very large part of the company’s future.

“Very large part” doesn’t even come close to describing it. By the middle of the century’s second decade, handheld devices and the operating system that ran them completely dominated Apple’s and the tech world’s attention, achieving the sort of celebrity status that the Mac and its operating system, influential as they had been, never even distantly approached.

Coming up, I look at what happened when a new user interface model arrived at Apple in The Case of the New Kid in Town, explore the mysteries of its evolution in The iOS and iPadOS Look and Feel Casebook, focus on the growing importance of screen edges in The iOS and iPadOS Edge Cases Casebook, and examine today’s editing interface in The iOS and iPadOS Touchy Text Casebook.
The Case of the New Kid in Town

The first iPhone was more of a promise than a fully fleshed-out product at its debut. By today’s standards, it was frustratingly limited. Even so, it offered enough of a breakthrough in ease of use and functionality for *Time* magazine to name it the *Invention of the Year*. Most importantly, it was wildly successful—and quickly copied by competitors. Its success, and the fierce competition it provoked, put the device on a developmental fast track at Apple, with the goal of keeping it one step ahead of competing devices. The pace has hardly slackened over the years, and users have been trying to adjust to the rapid pace of iOS device upgrades and enhancements ever since.

The iPhone and Its Early Limitations

Despite its very real limits, though, at first glance the iPhone that Jobs brandished that January morning seemed a polished product of sophisticated design, with a user interface that seemed both simple and powerful. The packaging emphasized the simplicity: when the first iPhones shipped the only user manual that came with it consisted of a small pamphlet of “Finger Tips” outlining the touch gestures you could use to control the device. That list of gestures was short: tap, hold, pinch in and out, swipe, flick. These were all simple gestures most users could easily learn and manage quickly.

At the time, no one had deployed sophisticated touchscreen interface devices to a mass market before, at least not in the numbers that Apple anticipated. Apple’s designers had had to build the interface for the iPhone on the basis of a relatively small corpus of research studies, carefully controlled internal testing, and some inspired guesswork—and do it all in secret.
It was a gamble: something that works really well for a small group of intelligent, highly educated technology workers can easily fall flat on its face when unleashed among a vastly diverse mass of paying customers. To succeed, the iPhone could not afford a repeat of the public shaming that the Newton, with its good but mockably imperfect handwriting recognition, had elicited. The iPhone, instead, needed to replicate the iPod experience, to offer an interface that a wide range of users could easily figure out and that was reliable within its limits.

Limits were the key: not a lot of gestures to learn, not a lot of apps to run—and least, not at first. The iPhone had been on the market for a year before any developers other than Apple could supply apps for the iPhone, and, unlike the Mac at that time, iPhone software would be sold only through an Apple-controlled App Store that had stringent rules about what kinds of apps, with what kinds of functionality, it would sell. Apple further reined in developer ambition by being the sole provider of the tools and code libraries that developers needed to develop iPhone apps, giving them access to only the functionality that Apple considered safe to deploy. The goal was to ensure that the iPhone was a reliable and easily accessible device: it may have been a revolutionary device, but it was a revolution that customers had to feel comfortable with.

Here’s a telling example of Apple’s cautious early approach to iPhone interface design: the first versions of the iPhone OS did not offer users any way to select text so that it could be copied and pasted elsewhere. It wasn’t that cut-and-paste was a computational mystery for Apple, but that Apple hadn’t yet developed a user interface for selecting text on a touchscreen that would be easy to use and feel natural. Apple wanted—needed—to make sure it got this crucial interface functionality right. And Apple eventually did get it right, though it took two years before cut and paste came to the iPhone (Figure 49). That Apple’s implementation was satisfactory can be seen in how closely other touch device platforms imitated Apple’s approach when it did show up.
The iOS and iPadOS Look and Feel Casebook

When the iPad made its first appearance, the more cynical among the digital punditry proclaimed that it was really nothing more than a big-screen iPhone. Superficially, they were correct: the iPad and iPhone used the same touchscreen technology, ran the same operating system, and had user interfaces that were almost indistinguishable. But the size discrepancy was significant in how the two types of devices would come to be used, just as the difference in size between a jewelry hammer and a sledge hammer imply quite different uses. And, over time, those different uses would cause the interfaces of iPhones and iPads to diverge, leading, eventually, to a divergence in their operating systems (see The Case of the Separated Siblings).

Still, the iOS and iPadOS user interfaces continue to have far more in common than not, and the mysteries they often present to users are generally similar and frequently identical.

Note: Even though Apple has now publicly named iPadOS as the operating system that drives its iPad product line, in the remainder of this chapter the phrase “iOS device” refers to all three Apple touchscreen product lines—iPhone, iPod touch, and iPad.

The Affordance That Hid in Plain Sight

As we learned in What Affordances Are, anything that an environment offers the user is an affordance. And the biggest affordance that iOS and iPadOS devices offer is the touch-sensitive screen that covers nearly all of the front surface of the devices. What few physical buttons they offer pale to insignificance by comparison with the screen, the physical component that dominates iOS and iPadOS device user interaction.
The Basic Gesture Inspection

The touch-sensitive screen, dubbed “Multi-Touch” by Apple, allows users to perform not only a variety of single finger gestures but multi-finger gestures. The basic gestures supported by iOS, even before it was called “iOS,” have remained relatively constant, with a handful of single-finger gestures and only one multi-finger gesture. These basic gestures, however, sometimes offer mysterious nuances. The gestures include:

- **Tap:** A single tap with a fingertip; it’s the gesture you use on an app’s icon to launch it, to select an item, or to press an onscreen button.

- **Double-tap:** Two quick taps with a fingertip. If you think it’s like a mouse double-click in macOS or the way to be sure you’ve *[put down a zombie]*, you’d be wrong: generally, you use this gesture to zoom in or out of an image.

- **Touch and hold:** This gesture elicits a menu or tool from the touched item. For example, a tap on editable text places an insertion point in the text, but a touch-and-hold gesture on the text evokes the Edit menu when you release the hold. This gesture can also cause an app to change modes: for example, when you touch and hold on a Home screen icon in iOS 13, the gesture first causes a menu of options related to that icon to appear *(Figure 51)*, but holding longer before releasing puts the Home screen into what Apple officially calls “jiggle mode,” at which point you can drag the icons around on the screen to rearrange them.
What’s beyond the edges of the screen has always been important in iOS over the years: the keyboard that slides up from below the screen’s bottom, the search field that slides down from the top, the Home screens that slide in from the left or right, and so on. But though the screen edges have always been interesting in terms of what lies beyond them, the physical screen edges have not loomed particularly large in Apple’s gestural interface repertoire until relatively recently, and it was because of another physical element: the Home button.

The Home Button That Vanished

Apple has never been fond of physical buttons. Aesthetic issues aside (and it is no secret that Steve Jobs and Jonny Ive liked clean, uncluttered designs), physical buttons add to a device’s cost of materials, and if they break they may require costly repairs. When the Mac came out, its mouse had but a single button at a time when any self-respecting pointing device had at least two. When the iPhone debuted, it, too, featured only one button on its face, its Home button, at a time when nearly every other mobile phone had a face full of buttons—some had full typewriter-style keyboards (with very tiny keys). Yet for Apple’s designers, even that lone iPhone button was one too many: on a small-screen device, every bit of screen space helps, and the space taken up by the Home button could be put to better use as extra screen space.

With the iPhone X, Apple did just that: the front of the device was almost all screen, with just a tiny notch at the top where the Face ID cameras and ear speaker lived. The Home button wasn’t moved to the device’s edges where the volume and sleep/wake buttons lived: it was removed completely. That change meant Apple had to offer other ways for users to perform the simple tasks that clicks or presses of the Home
button used to perform. In several cases, those other ways were new
gestures, and some of those gestures took advantage of the new edge-
to-edge iPhone display by using its edges as part of the gesture.

And as long as edge-based gestures were the new hotness, Apple gave
the iPad several of its own.

---

**The Great Swipe Hunt**

From the beginning the swipe in iOS has been a somewhat difficult-to-describe gesture. That’s because this touch-and-slide motion very much resembles two other basic gestures, the flick and the drag (see *The Basic Gesture Inspection*). It’s what each gesture is usually used for that sets them apart: a drag usually moves an object from one place on the screen to another (such as an app icon in jiggle mode), a flick usually scrolls through a view of something (such as a webpage), and a swipe usually reveals something previously hidden (such as a control).

**The Swipes That Switch**

Once the Home button was gone, Apple chose a revelatory swipe
gesture to replace the Home button clicks that had formerly been used to dismiss the Lock screen and reveal the Home screen.

This Home swipe gesture starts at the bottom edge of the screen (precisely where along the bottom edge doesn’t matter) and is a short upward swipe. To help you figure out that the bottom of the screen is an Important Place in the Home button-less world, iOS and iPadOS place a Home indicator line at the bottom of the screen (*Figure 62*).

*Figure 62*: The Home indicator at the iPhone screen’s bottom edge is where you swipe up to open the Lock screen.

---

Click here to buy the full 173-page “Apple Interface Mysteries” for only $9.99!
Although using mobile phones for texting was already quite popular when the iPhone first appeared, mobile phones of that era were far from text-handling powerhouses. Text entry, when it was available, mostly allowed you to create short text messages of a dozen or so words at a time, enter address information in contact lists, type in short (very short) details for calendar appointments, and generate other small snippets of alphanumeric text. The majority of mobile phones didn’t even offer a full text keyboard: you used their telephone keypads for text entry (tap the 2 key once for “A,” twice for “B,” and so on). Editing? Don’t bother; just backspace over your typos and retype: it’s not like you’re transcribing *À la recherche du temps perdu*, buddy!

The full, albeit virtual, keyboard available on the first iPhone provided text entry capabilities beyond what all but the top tier of business-oriented mobile phones, encrusted with tiny physical keyboards, could provide. That the iPhone lacked any form of basic text-editing capabilities was a sore point (cut, copy, paste, and undo were completely absent, as I’ve mentioned elsewhere), but it was not a very sore point, as pretty much no one else provided them either. Nonetheless, those niceties were looming, even though no one outside Apple knew when and how they would appear.

Finally, a text selection interface, along with cut, copy, paste, and undo, appeared in iPhone OS 3.0, released in June 2009 along with the iPhone 3GS. At that time, Apple was less than a year away from releasing the first iPad: *that* device, with a virtual keyboard large enough to touch-type on, cried out for more robust text editing capabilities. Were Apple to release the iPad with only the primitive text-entry capabilities that the original iPhone provided, it would be greeted with howls of derision. In a nutshell, iOS 3.0 provided iPhone users with text editing features because the secret still-in-development iPad
needed an operating system that had text editing features, and Apple needed to field test those features.

As we all know, those features worked well enough that they remained almost unchanged in iOS 3.2, which was released in April 2010 along with the iPad. iOS users on Apple multitouch devices both large and small could now cut, copy, and paste text with abandon, and they could undo edits as well, although the newly introduced technique of shaking the device to undo an action was less physically taxing—albeit no less comical—with an iPhone than with its much larger, much heavier, sibling.

And for many years those editing features remained almost unchanged, being good enough to support full blown word processors like Apple’s own Pages software. But lately, with the advent of iPadOS and Apple’s push to position its Pro model iPads in the liminal market space that lies between tablets and laptop computers, the text handling capabilities of both iOS and iPadOS have been supplemented, tweaked, and polished.

The Dancing Fingers Oscillation

Apple faced a big challenge developing an interface that could support the complexities of text-editing on a handheld touchscreen device: the human finger itself. The dexterous digit, our species’ built-in pointing device, although adept is big and clumsy and opaque, ill-designed to range accurately over lines of text and place an insertion point exactly where you’d like it.

An experiment: place the tip of your finger on a word in this ebook... say, right here ☞ word ☜...and ask yourself how much of that word you can see. Unless you’ve enlarged the text of this book significantly, or you have incredibly dainty fingers, chances are good that your fingertip covers the whole word. Wonderful as the finger is, it’s a blunt object where, for accurate insertion point placement, a fine point is needed.
Crime Fighting Tips

The Apple-created confusions and complexities catalogued in the preceding chapters are by no means exhaustive, but merely representative of how and, just as importantly, why the main ways you interact with Apple devices have become more complicated over the years. These days (and for many many of the days preceding these days) few people can claim with a straight face that Apple products are easy to use, let alone intuitive. To use them productively you have to do a considerable amount of learning—even if all you want is to browse a few webpages, send an email or text, take a picture, or make a phone call.

I can’t unravel every mystery you’ll stumble over in the Apple ecosystem, if for no other reason than that new ones pop up all the time. Your Mac, your iPhone, and your iPad mutate a little (and sometimes much more than a little) every time you upgrade their software. And skipping all upgrades is not really a solution either, as Adam Engst has explained. At best I can give you coping mechanisms, strategies for dealing with the inevitable changes and concomitant confusions that new and upgraded software and hardware bring.

In this chapter I take another look at the various takeaways I’ve sprinkled through this book and then I offer a few pointers toward resources you can consult.

Takeaway Retakes

The takeaways I offer following many of the topics in this book are short observations or bits of guidance related to the topics they follow. I collect the takeaways and say a little more about them here.
How Interfaces Happen

Early on, in *What Usability Is*, I provided the following observation:

**The Takeaway:** As random and arbitrary as some user interfaces can seem, most user interfaces don’t just happen. Lots of people, working hard and thinking hard, consciously and deliberately craft them. When interfaces fall short of perfection it’s seldom from want of trying. As Osgood Fielding III says at the conclusion of *Some Like It Hot*, “Nobody’s perfect!”

Hoo-boy! You should have seen the chain of comments in this book’s manuscript between Joe and me about this one. Both of us have been part of software development teams over the years, and we each have *very* decided opinions on whether interfaces “just happen” or not. Joe’s final comment to me was, “In my experience, a lot of bad UI happens because it’s designed by engineers without any usability training, oversight, or testing.” I cannot completely disagree with that, only on how much “a lot” is.

Yes, there are many mediocre developers out there, and many development teams that have neither a broad nor deep understanding of best user interface practices. Even a company as big and powerful as Apple, with the money and resources capable of luring and retaining the best of the best, has troubled teams, teams that may be overwhelmed by unrealistic development schedules or riven with strife and internal politics (very smart people can be just as fractious as the less gifted—and far more capable of sowing chaos). There are all sorts of ways perfection can elude Apple’s grasp.

Nonetheless, I still believe that most interfaces don’t just happen: attention is usually paid, although it is not always paid by those particularly skilled or interested in human interface methodologies or by those with an empathetic understanding of the people who must eventually use their creations. But in all but the most egregious cases of bad user interfaces, there is some underlying logic, some reason why the interface is presented in the manner that it has been.
About This Book

Thank you for purchasing this Take Control book. We hope you find it both useful and enjoyable to read. We welcome your comments.

Ebook Extras

You can access extras related to this ebook on the web. Once you’re on the ebook’s Take Control Extras page, you can:

- Download any available new version of the ebook for free, or buy a subsequent edition at a discount.
- Download various formats, including PDF, EPUB, and Mobipocket. (Learn about reading on mobile devices on our Device Advice page.)
- Read the ebook’s blog. You may find new tips or information, as well as a link to an author interview.
- Find out if we have any update plans for the ebook.

If you bought this ebook from the Take Control website, it has been automatically added to your account, where you can download it in other formats and access any future updates. However, if you bought this ebook elsewhere, you can add it to your account manually:

- If you already have a Take Control account, log in to your account, and then click the “access extras...” link above.
- If you don’t have a Take Control account, first make one by following the directions that appear when you click the “access extras...” link above. Then, once you are logged in to your new account, add your ebook by clicking the “access extras...” link a second time.

Note: If you try these directions and find that your device is incompatible with the Take Control website, contact us.

Click here to buy the full 173-page “Apple Interface Mysteries” for only $9.99!
About the Author

Michael E. Cohen lives on the southwest corner of the intersection of technology and the liberal arts. He’s taught writing and literature courses, programmed for the Deep Space Network, designed and implemented instructional word processors, produced interactive CD-ROMs, helped create the first commercial ebooks, developed and documented commercial software for an international corporation, written a novel, drawn a number of remarkably bad cartoons, and advised dozens of humanities professors on all matters digital.

As a young man, Michael set out to be a professional dilettante and, after years of struggle, succeeded beyond his wildest dreams.

Acknowledgments

I must thank this guy named Joe, who claims to be my publisher, for suffering through the long delays and scheduling stress that my project inflicted on him, and even more for providing useful feedback and acting as a sounding board. Additional thanks to my brother Norman, a very knowledgable long-time Apple user in both personal and professional settings, for reading early portions of the manuscript and
providing invaluable tips and sanity checks. Neither Joe nor Norman should be held responsible for any (if not many) errors in analysis I committed in this book.

About the Publisher

alt concepts inc., publisher of Take Control Books, is operated by Joe Kissell and Morgen Jahnke, who acquired the ebook series from TidBITS Publishing Inc.’s owners, Adam and Tonya Engst, in May 2017. Joe brings his decades of experience as author of more than 60 books on tech topics (including many popular Take Control titles) to his role as Publisher. Morgen’s professional background is in development work for nonprofit organizations, and she employs those skills as Director of Marketing and Publicity. Joe and Morgen live in San Diego with their two children and their cat.

Credits

• Editor and Publisher: Joe Kissell
• Cover design: Sam Schick of Neversink
• Logo design: Geoff Allen of FUN is OK

More Take Control Books

This is but one of many Take Control titles! We have books that cover a wide range of technology topics, with extra emphasis on the Mac and other Apple products.

You can buy Take Control books from the Take Control online catalog as well as from venues such as Amazon and the Apple Books Store. But it’s a better user experience and our authors earn more when you buy directly from us. Just saying...

Our ebooks are available in three popular formats: PDF, EPUB, and the Kindle’s Mobipocket. All are DRM-free.
**Copyright and Fine Print**

*Apple Interface Mysteries*
ISBN: 978-1-947282-54-4
Copyright © 2020, Michael E. Cohen. All rights reserved.

[alt concepts inc.](http://www.altconcepts.com), 4142 Adams Ave. #103-619, San Diego CA 92116, USA

**Why Take Control?** We designed Take Control electronic books to help readers regain a measure of control in an oftentimes out-of-control universe. With Take Control, we also work to streamline the publication process so that information about quickly changing technical topics can be published while it’s still relevant and accurate.

**Our books are DRM-free:** This ebook doesn’t use digital rights management in any way because DRM makes life harder for everyone. So we ask a favor of our readers. If you want to share your copy of this ebook with a friend, please do so as you would a physical book, meaning that if your friend uses it regularly, they should buy a copy. Your support makes it possible for future Take Control ebooks to hit the internet long before you’d find the same information in a printed book. Plus, if you buy the ebook, you’re entitled to any free updates that become available.

**Remember the trees!** You have our permission to make a single print copy of this ebook for personal use, if you must. Please reference this page if a print service refuses to print the ebook for copyright reasons.

**Caveat lector:** Although the author and alt concepts inc. have made a reasonable effort to ensure the accuracy of the information herein, they assume no responsibility for errors or omissions. The information in this book is distributed “As Is,” without warranty of any kind. Neither alt concepts inc. nor the author shall be liable to any person or entity for any special, indirect, incidental, or consequential damages, including without limitation lost revenues or lost profits, that may result (or that are alleged to result) from the use of these materials. In other words, use this information at your own risk.

**It’s just a name:** Many of the designations in this ebook used to distinguish products and services are claimed as trademarks or service marks. Any trademarks, service marks, product names, or named features that appear in this title are assumed to be the property of their respective owners. All product names and services are used in an editorial fashion only, with no intention of infringement. No such use, or the use of any trade name, is meant to convey endorsement or other affiliation with this title.

**We aren’t Apple:** This title is an independent publication and has not been authorized, sponsored, or otherwise approved by Apple Inc. Because of the nature of this title, it uses terms that are registered trademarks or service marks of Apple Inc. If you’re into that sort of thing, you can view a [complete list](http://www.altconcepts.com) of Apple Inc.’s registered trademarks and service marks.

Click here to buy the full 173-page “Apple Interface Mysteries” for only $9.99!