iPhone IN ACTION

Introduction to Web and SDK Development

SAMPLE CHAPTER

Christopher Allen
Shannon Appelcline

MANNING

iPhone in Action

Introduction to Web and SDK Development

by Christopher Allen
and Shannon Appelcline

Chapter 1

brief contents

Part 1

Introducing iPhone programming

Apple's iPhone is more than just a new programming platform; it's an entirely new way to think about mobile technologies. Part 1 of this book gets your feet wet by explaining how the iPhone differs from its predecessors.

We'll start off in chapter 1 with a look at the iPhone itself, then follow up in chapter 2 by describing the two ways you can program for the iPhone: by creating web apps and by using the iPhone SDK.

1 Introducing the iPhone

This chapter covers

- Understanding Apple's iPhone technology
- Examining the iPhone's specifications
- Highlighting what makes the iPhone unique

In the 1980s Apple Computer was the leading innovator in the computer business. Their 1984 Macintosh computer revolutionized personal computing and desktop publishing alike. But by the 1990s the company had begun to fade; it was depending on its loyal user base and its successes in the past rather than creating the newest cutting-edge technology.

That changed again in 1996 when founder Steve Jobs returned to the fold. Two years later he produced the first candy-colored iMac, a computer that walked the line between computing device, pop culture, and fashion statement. It was just the first of several innovations under Jobs' watch, the most notable of which was probably 2001's iPod. The iPod was a masterpiece of portable design. It highlighted a simple and beautiful interface, giving users access to thousands of songs that they could carry with them all the time. But the iPod just whetted the public's appetite for more.

By 2006 rumors and speculation were rumbling across the internet concerning Apple's next major innovation: an iPod-like mobile phone that would eventually be called the iPhone. Given Apple's twenty-first century record of technological innovation and superb user design, the iPhone offered a new hope. It promised a new look at the entire cellular phone industry and the possibility of improved technology that wouldn't be afraid to strike out in bold new directions.

Apple acknowledged that they were working on an iPhone in early 2007. When they previewed their technology, it became increasingly obvious that the iPhone would be something new and different. Excitement grew at a fever pitch. On the release date—June 29, 2007—people camped outside Apple stores. Huge lines stretched throughout the day as people competed to be among the first to own what can only be called a *smarterphone*, the first of a new generation of user-friendly mobile technology.

When users began to try out their new iPhones, the excitement only mounted. The iPhone was easy to use and it provided numerous bells and whistles, from stock and weather reports to always-on internet access. Sales reflected the frenzied interest. Apple sold 270,000 iPhones in two days and topped a million units in just a month and a half. Now, a year and a half after the initial release, interest in the iPhone continues to grow. Apple's July 11, 2008, release of the new 3G iPhone and its public deployment of the iPhone software development kit (SDK) promise to multiply the iPhone's success in the future, with even higher numbers of iPhone sales predicted for 2009 and beyond. The 3G managed to hit a million units sold in just three days. We're atop a new technological wave, and it has yet to crest.

But what are the technologies that made the iPhone a hit, and how can you take advantage of them as an iPhone programmer? That will be the topic of this first chapter, where we'll not only look at the core specifications of the iPhone but also discuss the six unique innovations that will make developing for the iPhone a totally new experience.

1.1 iPhone core specifications

The iPhone is more than a simple cell phone and more than a smartphone like the ones that have allowed limited internet access and other functionality over the last several years. As we've already said, it's a *smarterphone*. If the iPod is any indication of market trends, the iPhone will be the first of a whole new generation of devices but will simultaneously stay the preeminent leader in the field because of Apple's powerful brand recognition and its consistent record of innovation.

Technically, the iPhone exists in two largely similar versions: the 2007 original release and the 2008 3G release. Each is a 4.7- or 4.8-ounce computing device. Each contains a 620 MHz ARM CPU that has been underclocked to improve battery performance and reduce heat. Each includes 128 MB of dynamic RAM (DRAM), and from 4 to 16 GB of Flash memory. The primary differences between the two devices center on the global positioning system (GPS) and networking, topics we'll return to shortly.

Programmatically, the iPhone is built on Apple's OS X, which is itself built on top of Unix. Xcode, the same development environment that's used to write code for the Macintosh, is the core of native programming for the device. Putting these two

elements together reveals a mature development and runtime environment of the sort that hasn't been seen on most other cell phones (with the possible exception of Windows Mobile) and that upcoming smarterphone technologies won't be able to rival for years.

> **The iPod Touch**
>
> A few months after the release of the original iPhone, Apple updated their iPod line with the iPod Touch. This was a new iPod version built on iPhone technology. Like the iPhone, it uses a 480x320 multi-touch screen and supports a mobile variant of Safari.
>
> However, the iPod Touch is *not* a phone. The original version didn't have any other telephonic apparatus, nor did it include other iPhone features such as a camera. The year 2008 saw the release of a new version of the iPod Touch that included an external speaker and volume controls missing from the original 2007 model. Because of its lack of cellular connectivity, the iPod Touch can only access the internet through local-area wireless connections.
>
> The developer advice in this book will largely apply to the iPod Touch as well, though we won't specifically refer to that device.

However, these general specs tell only part of the story. By looking deeper into the iPhone's input and output, its network, and its other capabilities, you'll discover what makes the iPhone a truly innovative computing platform.

1.1.1 iPhone input and output specifications

Both the input and the output capabilities of the iPhone feature cutting-edge functionality that will determine how developers program for the platform. We're going to provide an overview of the technical specifications here; later in this chapter we'll start looking at the iPhone's most unique innovations, and then return for a more in-depth look at the input and output.

The iPhone's input is conducted through a multi-touch-capable capacitive touchscreen. There is no need for a stylus or other tool. Instead, a user literally taps on the screen with one or more fingers.

The iPhone's visual output is centered on a 3.5" 480x320-pixel screen. That's a larger screen than has been seen on most cell phones to date, a fact that makes the iPhone's small overall size that much more surprising. The device is literally almost all screen. The iPhone can be flipped to display either in portrait or landscape mode, meaning that it can offer either a 480-pixel-wide or a 480-pixel-tall screen.

The iPhone's output also supports a variety of media, all at the high level that you'd expect from the designers of the iPod. Music in a number of formats—including AAC (Advanced Audio Coding), AIFF (Audio Interchange File Format), Apple Lossless, Audible, MP3, and WAV—is supported, as well as MPEG4 videos. Generally, an iPhone delivers CD-quality audio and high frame rate video.

Although users will load most of their audio and video straight from their computer, the iPhone can play streams at a recommended 900 kbps over wi-fi, but that can be pushed much higher on a good network. Multiple streaming rates always choose the optimal method for the current network interface—which brings us to the question of the iPhone's networking capabilities.

1.1.2 iPhone network specifications

The iPhone offers two methods of wireless network connectivity: local area and wide area.

The iPhone's preferred method of connectivity is through a local-area wireless network. It can use the 802.11g protocol to link up to any nearby wi-fi network (provided you have the permissions to do so). This can provide local connections at high speeds of up to 54 megabits per second (Mbit/s), thus making a network's link to the internet the most likely source of speed limits, not the iPhone itself. Everything has been done to make local-area connectivity as simple to use as possible. Passwords and other connection details are saved on the iPhone, which automatically reconnects to a known network whenever it can. Switches to and from local wi-fi networks are largely transparent and can happen in the middle of internet sessions.

The original iPhone uses the EDGE network for wide-area wireless connectivity, falling back on this network whenever local-area wireless access isn't available. The EDGE network supports speeds up to 220 kilobits per second (kbit/s). Compared to old-style modems, which were accessing the early internet just 15 years ago, this is quite fast, but compared to broadband connectivity it's not that good. Although the original iPhones have *already* been phased out, millions of users are still using them, and thus EDGE network speed remains relevant.

The 3G iPhone supports the third-generation of mobile phone standards, which are well developed in Europe but just emerging in the United States. Network speed standards for 3G are loose, with stationary transfer speeds estimated as low as 384 kbit/s or as high as several Mbit/s. A 3G connection should generally be noticeably quicker than EDGE but still not as fast as a local-area network. In addition, 3G iPhones may drop back to EDGE connectivity if there's insufficient 3G coverage in an area.

These network specifications will place the first constraints on your iPhone web development (and will be of relevance to SDK programs that access the internet as well). If you're working in a corporate environment where everyone will be accessing your apps through a companywide wi-fi, you probably don't need to worry that much about how latency could affect your application. If you're creating iPhone web pages for wider use, however, you have to presume that some percentage of your iPhone users will be accessing them via a wide-area wireless network. This should encourage developers to fall back on lessons learned in the 1990s. Web pages should be smaller and use clean style sheets and carefully created images; data should be downloaded to the iPhone using Ajax or other technologies that allow for sporadic access to small bits of data.

The web vs. the SDK

Throughout the book we're going to talk about two major categories of programming for the iPhone: web development and SDK programming.

Web development involves the creation of web pages that work well on the iPhone. These pages use standard web technologies such as HTML, Cascading Style Sheets (CSS), JavaScript, PHP, Ruby on Rails, and Python; iPhone-specific technologies such as the WebKit, iUI, and Canvas; and iPhone-specific tools like Dashcode.

SDK programming involves the design of programs that run natively on the iPhone. These programs are written in Objective-C, compiled in Xcode, and then deployed through the iPhone App Store.

We'll compare the two programming methods in the next chapter; then we'll dedicate part 2 of this book to learning all about iPhone web development and parts 3 and 4 to digging into Apple's iPhone SDK.

Thus far, we've discussed iPhone specifications that are relevant to both web and SDK development. However, there's one additional element that's clearly web only: the browser.

1.1.3 iPhone browser specifications

The iPhone's browser is a mobile version of Apple's Safari. It's a full-fledged desktop-grade browser with access to DOM, CSS, and JavaScript. However, it doesn't have access to some client-side third-party software that you might consider vital to your web page's display.

The two most-used third-party software packages that aren't available natively to the iPhone are Flash and Java. There was some discussion of a Java release in 2008, but the SDK's restriction against downloads seems to have put that effort on hold. We'll talk about these and other "missing technologies" more in chapter 3.

Beyond listing what's available for the iPhone Safari browser (and what's not), we'll also note that it works in some unique ways. There are numerous small changes that optimize Safari for the iPhone. For example, rather than Safari's standard tabbed browsing, individual "tabs" appear as separate windows that a user can move between as if they were individual pages.

iPhone's Safari also features unique "chrome," which is its rendition of toolbars. These gray bars appear at the top and bottom of every iPhone web screen. The chrome at the top of each page shows the current URL and features icons for bookmarks and reloading; we'll investigate how to hide this chrome when we look at iPhone optimized web development in chapter 3. The chrome at the bottom contains additional icons for moving around web pages and tabs. It's a permanent fixture on iPhone web pages. This iPhone chrome is more noticeable than similar bars and buttons on a desktop browser because of the iPhone's small screen size.

Having discussed the general capabilities of the iPhone—its input, its output, its network, and its browser—we've hit all of the major elements. But the iPhone also has some additional hardware features that are worthy of specific note.

1.1.4 Other iPhone hardware features

Cell phones feature numerous hardware gadgets—of which a camera is the most ubiquitous. The iPhone includes all of the cell phone standards, but also some neat new elements, as outlined in table 1.1.

Table 1.1 The iPhone is full of gadgets, some of them pretty standard for a modern cell phone, but some more unique.

Gadget	Notes
Accelerometers	The iPhone contains three accelerometers. Their prime use is to detect an orientation change with relation to gravity—which is to say they sense when the iPhone is rotated from portrait to landscape mode or back. However, they can also be used to approximately map an iPhone's movement through three-dimensional space. Could this make the iPhone the next Wii?
Bluetooth	This standard protocol for modern cell phones allows access to wireless devices. The iPhone uses the Bluetooth 2.0+EDR protocol. Enhanced Data Rate (EDR) allows for a transmission rate about triple that of older versions of Bluetooth (allowing for a 3.0 Mbit/s signaling rate and a practical data transfer rate of 2.1 Mbit/s).
Camera	Another de facto requirement for a modern cell phone. The iPhone's camera is 2.0 megapixel.
GPS	The original iPhone doesn't support real GPS, but instead offers the next best thing, "peer-to-peer location detection." This faux GPS triangulates based on the relative locations of cell phone towers and wi-fi networks for which real GPS data exists, and then extrapolates the user's location based on that. Unfortunately the accuracy can vary dramatically from a potential radius of several miles to several blocks; still, it's better than no GPS at all. The 3G iPhone includes a true Assisted GPS (A-GPS), which supplements normal GPS service with cell network information. Although there is a difference in accuracy between the two types of GPSs, they can both be accessed through the iPhone SDK using the same interface.

Of these hardware features, the ones that really stand out are the accelerometers and the GPS, which are not the sort of things commonly available to cell phone programmers. As you'll see, they spotlight two of the elements that make the iPhone unique: orientation awareness and location awareness. However, before we fully explore the iPhone's unique features, it's useful to put the device in perspective by comparing the iPhone to the mobile state of the art.

1.2 How the iPhone compares to the industry

Although the iPhone is an innovative new technology, it also serves as a part of a stream of mobile development that's been ongoing for decades. Understanding the iPhone's place within the industry can help us to better understand how it's differentiated from the rest of the pack.

1.2.1 The physical comparison

Physically, the iPhone is the sort of stunningly beautiful device that you'd expect from Apple. As we already said, it's almost all screen, highlighting Apple's ability to transform expectations for their electronic devices.

More specifically, the iPhone has a much larger screen than most of the last-generation cell phones, which tended to run from 320x240 pixels to 320x320 pixels and thus had as few as half as many pixels to play with as the iPhone. Although they had keyboards that were comparable with the iPhone's on-screen keyboard, their mousing methods were often based around styluses or tiny trackballs or, worse, scrolling wheels.

We expect other cell phones to start catching up with the iPhone's physical specs pretty quickly, but in the meantime Apple has used those specs to create a totally new cell phone experience—starting with its improved internet experience.

1.2.2 Competitive internet viewing

When compared to its last-generation competitors, the iPhone produces an internet experience that is more usable, better integrated, and more constant than the standard mobile experience.

The improvements in usability stem from the innovative specifications that we've already seen for input, output, and networking. On the input side, you no longer have to use a last-generation scrolling wheel to painfully pick your way through links up and down a page. On the output side, pages are displayed cleanly and crisply without being broken into segments, thus allowing for a faster, more pleasant web experience. Finally, for networking, you have the relatively good speed of the EDGE or 3G network combined with the ability to use lightning-fast local-area networks whenever possible. When compared to last-generation phones plagued by molasses-like internet connections, the change is striking.

With such a strong foundation, Apple took the next step and integrated the internet into the whole iPhone experience in a way that last-generation cell phones failed to do. The iPhone includes a variety of standard programs such as a YouTube interface, a stock program, a maps program, and a weather program that all provide seamless, automatic access to the internet. In addition, the SDK provides simple access to the internet for original applications.

All this functionality is supported by a constancy of internet access that is unlike anything the smartphone industry has ever seen. Supplementing its wi-fi access, an iPhone can access the internet through cheap add-on data plans. These plans allow for unlimited data transfer via the web and email. Thus, users never have to think about the cost of browsing the web. The end result is an always-on internet that, as we'll see, is another of the elements that makes the iPhone truly unique.

The Apple iPhone has brought mobile internet browsing out of the closet, a fact that is going to result in notable changes to current mobile web standards.

1.2.3 Mobile web standards

Prior to the release of the iPhone, a number of web standards were being developed for smartphones. The .mobi top-level domain was launched in 2006, built on the Wireless Application Protocol (WAP) and the Wireless Markup Language (WML) standard for cut-down, mobile HTML. In addition, the W3C Mobile Web Initiative has begun work on standards such as mobileOK (which is meant to highlight mobile best practices).

It is our belief that the mobile standards—and even the .mobi domain—are for the most part irrelevant when applied to the iPhone. We believe so because the iPhone provides a fully featured desktop-class browser and has vastly improved input, output, and networking capabilities. There *are* best practices for developing on the iPhone, and we'll talk about some of them in upcoming chapters, but they're not the same best practices required for leading-edge designs prior to 2007. As more smarterphones appear, we believe that the mobile standards being worked on now will quickly become entirely obsolete.

This is not to say, however, that the iPhone is without limitations. It does not and cannot provide the same experience as a desktop display, a keyboard, and a mouse. New mobile standards for smarterphones will exist; they'll simply be different from those developed today.

Before completing our comparison of the iPhone to the rest of the industry, it's important to note that the vastly improved and integrated internet access of the iPhone is only part of the story.

1.2.4 The rest of the story

In 2008 Apple released the next major element in the iPhone revolution, the SDK, a developer's toolkit that allows programmers to create their own iPhone applications. Prior to the release of the SDK, most cell phone development kits were proprietary and highly specialized. The open release of the SDK could revolutionize the cell phone industry as much as the iPhone's web browsing experience already has.

Even that's not the whole story. The iPhone is an innovative product, top to bottom. To further highlight how it's grown beyond the bounds of the last-generation cell phone industry, we've identified six elements that make the iPhone truly unique.

1.3 How the iPhone is unique

The iPhone's core uniqueness goes far beyond its powerful web browser and its tightly integrated web functionality. Its unique physical form and the decisions embedded in its software also make the device a breakthrough in cell phone technology. Six core ideas—most of which we've already hinted at—speak to the iPhone's innovation. Understanding these elements (summarized in table 1.2) will help you in whatever type of development you're planning.

The idea of an *always-on internet* is something we already touched on earlier. However what's notable is how successful Apple has been in pushing this idea. Huge data transfer rates show that iPhone users are indeed *always-on*. In Europe, T-Mobile reported that their iPhone users transferred 30 times as much data as their regular

Table 1.2 The iPhone has a number of unique physical and programmatic elements that should affect any development on the platform.

Unique Element	Summary
Always-on internet	A well-integrated, constant internet experience
Power consciousness	A device that you can use all day
Location-aware	A device that knows where it is
Orientation-aware	A device that detects movements in space
Innovative input	Finger-based mousing
Innovative output	A high-quality scalable screen

users. Google has also shown a notable uptick among iPhone users, who are 50 times more likely to conduct a search than the average internet user. Looking at overall stats, the iPhone's mobile Safari has already become the top mobile browser in the United States and is quickly moving up in the international market as well. Anecdotal evidence is consistent, as friends talk about how an iPhone user is likely at any time to grab his or her iPhone to look up a word in Webster's or a topic in Wikipedia, showing off how the iPhone has become the encyclopedia of the 21st century for its users.

When Apple announced the iPhone, they highlighted its *power consciousness*. Users should be able to use their iPhone all day, whether they're talking, viewing the web, or running native applications. Despite the higher energy costs of the 3G network, the newest iPhone still supports 5 hours of talking or 5–6 hours of web browsing. Power-saving tricks are built deeply into the iPhone. For example, have you noticed that whenever you put your iPhone up to your ear, the screen goes black to conserve power? And that it comes back on as soon as you move the iPhone away from your ear? Power savings have also been built into the SDK, limiting some functionality such as the ability to run multiple programs simultaneously in order to make sure that a user's iPhone remains functional throughout the day.

Thanks to its GPS (true or faux), an iPhone is *location aware*. It can figure out where a user is, and developers can design programs to take advantage of this knowledge. To preserve users' privacy, however, Apple has limited what exactly programs can do with that knowledge.

Just as an iPhone is knowledgeable of large-scale location changes, it also recognizes small-scale movements, making it *orientation aware*. As we've already noted, this is thanks to three accelerometers within the iPhone. They don't *just* detect orientation; they can also be used to measure gravity and movement. Although some of this functionality isn't available to web apps, sophisticated input can be accessed by SDK programs.

Finally we come to the iPhone's *innovative input* and *output*. Thanks to a multi-touch screen and a uniquely scaled screen resolution, the iPhone provides a different interactive experience from last-generation cell phones, so much so that we've reserved an entire section for its discussion.

1.4 Understanding iPhone input and output

Although an iPhone has a native screen resolution of 480x320 pixels, web viewers won't see web pages laid out at that resolution. An iPhone allows a user to touch and tap around pages in a way somewhat similar to mousing, but it provides notable differences from a mouse interface.

These differences highlight the final notable elements in the story of what makes the iPhone unique.

1.4.1 Output and iPhone viewport

When using the iPhone for most purposes, you may note that it has a 480x320 screen that displays very clearly. This is not a far cry from the 640x480 video displays common on desktop computers in the late 1980s, albeit with more colors and crispness than those early EGA and VGA displays. Thus, as we mentioned when discussing the slower speeds of the wide-area network, we can again fall back on the lessons of the past when developing for the iPhone.

The iPhone's display becomes interesting when it's used to view web pages, because the 480x320 display doesn't show web pages at that size. Instead, by default a user looks at a web page that has been rendered at a resolution of 980 pixels (with a few exceptions, as we'll note when talking about web development). In other words, it's as if users pulled a web browser up on their computer screen that was 980 pixels wide, and then scaled it down by a factor of either 2:1 or 3:1—depending on the orientation of the iPhone—to display at either 480 or 320 pixels wide.

This scaled view is what the iPhone calls a "viewport." As you'll see, viewport size can be set by hand as part of a web page design, forcing a page to scale either more or less when it's translated onto the iPhone. However, for any web page without an explicit viewport command, the 980-pixel size is the default.

Realizing that most pages will scale by a factor of at least two is vital to understanding how web pages will look on an iPhone. In short, everything will be really, really small. As a result, good web development for the iPhone depends on ensuring that words and pictures appear at a reasonable size despite the scaling. We'll talk about how to do that using the viewport command, CSS tricks, and other methods in chapter 3.

And for SDK developers: note this is an issue for you as well, since the SDK's `UIWebView` class scales the screen just like mobile Safari does. We'll see the first example of this in chapter 11.

1.4.2 Output and orientations

We need to consider one other important element when thinking about the iPhone output: its ability to display in two different orientations, 480x320 or 320x480. Each orientation has its own advantages. The portrait orientation is great for listings, while the landscape orientation is often easier to read.

Each of these orientations also shows off the iPhone's "chrome" in a different way. This chrome will vary from one SDK program to another, but it's consistent when view-

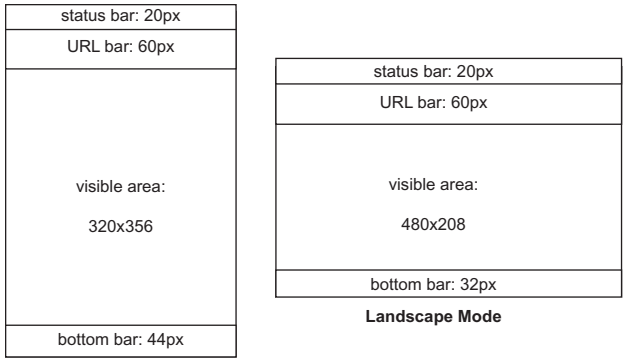

Figure 1.1 The iPhone supports two dramatically different views, landscape and portrait. Choosing between them is not just a question of which is easier to read, but also requires thinking about how much of each view is taken up by toolbars and other chrome. Mobile Safari is used here as an example of how much room the chrome takes up in each display.

ing web pages in Safari, and thus we can use the latter as an example of orientation's impact on chrome, as shown in figure 1.1.

One of the interesting facts shown by this picture is that the web chrome takes up a larger percentage of the iPhone screen in the landscape mode than in the portrait mode. This is summarized in table 1.3.

Mode	Chrome % with URL	Chrome % without URL
Portrait	26%	13%
Landscape	35%	16%

Table 1.3 Depending on an iPhone's orientation, you'll have different amounts of screen real estate available.

The difference between the orientations isn't nearly as bad without the URL bar, which scrolls off the top of the screen as users move downward, but when users first call up a web page on the iPhone in landscape mode, they'll only get to see a small percentage of it. You'll see similar issues in your SDK development too, particularly if you're creating large toolbars for your applications.

Despite this limitation of landscape mode, many of the best applications will likely shine in that layout, as the built-in YouTube application shows.

With discussions of viewports and orientations out of the way, we've highlighted the most important unique elements of the iPhone output, but its input may be even more innovative.

1.4.3 Input and iPhone mousing

As already noted, the iPhone uses a multi-touch-capable capacitive touch screen. Users access the iPhone by tapping around with their finger. This works very differently from a mouse.

It's perhaps most important to say, simply, that *the finger is not a mouse*. Generally a finger is going to be larger and less accurate than a more traditional pointing device. This disallows certain traditional types of UI that depend on very precise selection. For

example, there are no scroll bars on the iPhone. Selecting a scroll bar with a "fat finger" would either be an exercise in frustration or would require a huge scroll bar that would take up a lot of the iPhone's precious screen real estate. Apple solved this problem by allowing users to tap anywhere on an iPhone screen, then "flick" in a specific direction to cause scrolling.

Another interesting element of the touchscreen is shown off by the fact that *the finger is not singular*. Recall that the iPhone's touchscreen is *multi*-touch. This allows users to manipulate the iPhone with multi-finger "gestures." The "pinch" zooming of the iPhone is one such example. To zoom into a page, you tap two fingers on a page and then push them apart, while to zoom in you similarly push them together.

Finally, *the finger is not persistent*. A mouse pointer is always on the display, but the same isn't true for a finger, which can tap here and there without going anywhere in between. As you'll see, this causes issues with some traditional web techniques that depend on a mouse pointer moving across the screen. It also provides limitations that might be seen throughout SDK programs. For example, there's no standard for cut and paste, a pretty ubiquitous feature for any computer produced in the last couple of decades.

Besides resulting in some changes to existing interfaces, the iPhone's unique input interface also introduces a number of new touches (one-fingered input) and gestures (two-fingered input), as described in table 1.4.

Table 1.4 iPhone touches and gestures allow you to accept user input in new ways.

Input	Type	Summary
Bubble	Touch	Touch and hold. Pops up an info bubble on clickable elements.
Flick	Touch	Touch and flick. Scrolls page.
Flick, Two-Finger	Gesture	Touch and flick with two fingers. Scrolls scrollable element.
Pinch	Gesture	Move fingers in relation to each other. Zooms in or out.
Tap	Touch	A single tap. Selects.
Tap, Double	Touch	A double tap. Zooms a column.

When you're designing with the SDK, many of the nuances of finger mousing will already be taken care of for you. Standard controls will be optimized for finger use, and you'll have access only to the events that actually work on the iPhone. Chapter 14 explains how to use touches, events, and actions in the SDK. Even though some things will be taken care of for you, as an SDK developer you'll still need to change your way of thinking about input to better support the new device.

When you're developing for the web, you'll have to be even more careful. We'll return to some of the ways that you'll have to change your web designs to account for finger mousing in chapter 3. You'll also have to think about one other factor: internet standards. The web currently doesn't have any paradigms for flicks and other gestures,

and thus the events related to them are going to be totally new. In chapter 4 you'll meet some brand-new iPhone-specific events, but they're just the tip of the iceberg and it might be years before internet standards groups properly account for them.

1.5 Summary

This concludes our overview of the iPhone. Our main goals in this chapter were to investigate how the iPhone differs from other devices, to discover how it's unique on its own, and to learn how those changes might affect development work.

Based on what we've seen thus far, our biggest constraints in development will be the potentially slow network, the relatively small size of the iPhone screen, and the entirely unique input interface. Although the first two are common issues for other networked cell phones, the third is not.

On the other hand, you've also seen the possibilities of many unique features, such as the iPhone's orientation and location awareness, though you won't get to work with these functions until we discuss the SDK.

In the next chapter we'll look at more of the differences between web development and the SDK so that you can better choose which of them to use for any individual development project.

iPHONE DEVELOPMENT

iPhone IN ACTION

Christopher Allen and Shannon Appelcline

Free ebook
SEE INSERT

The iPhone explodes old ideas of a cell phone. Its native SDK offers a remarkable range of features including easy-to-build graphical objects, a unique navigation system, and a built-in database, all on a location-knowledgeable device. Websites and web apps can now behave like native iPhone apps, with great network integration.

iPhone in Action is an in-depth introduction to both native and web programming for the iPhone. You'll learn how to turn your web pages into compelling iPhone web apps using WebKit, iUI, and Canvas. The authors also take you step by step into more complex Objective-C programming. They help you master the iPhone SDK including its UI and features like accelerometers, GPS, the Address Book, SQLite, and many more. Using Apple's standard tools like Dashcode, Xcode, and Interface Builder, you'll learn how to best use both approaches: iPhone web and SDK programming.

This book is intended as an introduction to its topics. Proficiency with C, Cocoa, or Objective-C is helpful but not required.

What's Inside

- A comprehensive tutorial for iPhone programming
- Web development, the SDK, and hybrid coding
- Over 60 web, Dashcode, and SDK examples

"*The* entry to the world of iPhone."
—Aiden Montgomery, Wile Ltd.

"If you're new to iPhone development, this is *your* book!"
—Larry C. Whipple
Mobile Productivity, Inc.

"Get this book. It's pure gold."
—Martijn Dashorst
Author of *Wicket in Action*

"The quick & easy guide."
—Premkumar Rajendran
HCL Technologies

"The only book on iPhone development I will ever need."
—Rama Krishna Vavilala, Author of *ASP.NET AJAX in Action*

About the Authors

A technologist and entrepreneur, **Christopher Allen** is a popular speaker, co-founder of iPhoneDevCamp, and host of its Hack-a-thon. iPhone consultant **Shannon Appelcline** is a writer and programmer who develops social web software and has written numerous books.

For online access to the authors, code samples, and a free ebook for owners of this book, go to www.manning.com/iPhoneinAction

ISBN-13: 978-1933988863
ISBN-10: 193398886X
53999
9 781933 988863

MANNING US/CAN $39.99 [INCLUDING EBOOK]