Chapter 1

What Makes a Killer iPad App

In This Chapter

- Figuring out what makes an insanely great iPad application
- Discovering the features of the iPad that can inspire you
- Understanding Apple's expectations for iPad applications
- Making a plan for developing iPad software

Douglas Adams, in the bestseller *The Hitchhiker's Guide to the Galaxy* (conceived in 1971 and published in 1979), introduced the idea of a handy travel guide that looked "rather like a largish electronic calculator," with a hundred tiny flat press buttons and a screen "on which any one of a million 'pages' could be summoned at a moment's notice. It looked insanely complicated, and this is one of the reasons why the snug plastic cover it fitted into had the words DON'T PANIC printed on it in large friendly letters." According to Adams, this guide was published in this form because "if it were printed in normal book form, an interstellar hitchhiker would require several inconveniently large buildings to carry it around in."

The iPad is a hitchhiker's dream come true, and its users don't even have any reason to panic. The only "insanely complicated" part of the iPad experience may be trying to develop a killer app that best exemplifies the iPad's features, but that's why I think this book should have DON'T PANIC printed on its cover — it takes you through the entire process of imagining, creating, developing, testing, and distributing your iPad app. And in this chapter, I talk about what would make that app a killer app.

As you already know, the iPad is a new category of device — located somewhere between a Mac laptop and an iPod touch or iPhone in terms of its capabilities — that evolved from the iPhone design and uses the iPhone Operating System (OS).

The iPad already runs the 140,000+ iPhone apps in the Apple App Store with either pixel-for-pixel accuracy in a black box in the center of the display, or scaled up to full screen (which is done on the fly by doubling the pixels). The

App Store is loaded with travel and digital media apps, so you know already that the iPad as a "Hitchhiker's Guide" is not a fantasy. You may think it a fantasy that you could develop an iPad app in less than two months, starting from where you are now, with no iPad programming experience. But you *can* — the only question is whether you can make a *great* app, or even a *killer* app. To do that, you need to look at what it takes for an iPad app to be truly great.

Figuring Out What Makes a Great iPad Application

You use the same iPhone developer kit, and much of the same code, to develop iPad applications, and the iPad runs the same operating system as the iPhone, but the iPad is a bigger device with more horsepower and a larger display, as shown in Figure 1-1.

For many iPhone app developers, the iPad's larger display alone changes everything. Apple demonstrated exactly how *far* things have changed when the company demonstrated the iWork suite of productivity tools (Keynote for presentations, Numbers for spreadsheets, and Pages for word processing and page formatting) on the iPad, which would be unthinkable for today's iPhone.

Figure 1-1: The iPad runs the iPhone OS (left) and offers a larger display to show content such as a newspaper (right).

The biggest challenge in making a killer app for the iPad is to design for the iPad *experience,* and one reason why the iPad offers such a better experience than any Windows netbook or tablet computer is its sex appeal (which for many apps can mean *more excellent content* and *finer style*). For example, according to Douglas Adams, the Encyclopedia Galactica describes *alcohol* as "a colorless volatile liquid formed by the fermentation of sugars" and also notes "its intoxicating effect on certain carbon-based life forms." On the other hand, The Hitchhiker's Guide to the Galaxy not only tells you what *alcohol* is, it says "the best drink in existence is the Pan Galactic Gargle Blaster," describes its effect as "like having your brains smashed out by a slice of lemon wrapped round a large gold brick," tells you which planets have bars that offer it and at what prices, and then shows you how to mix one yourself. As Adams points out, "The Hitchhiker's Guide to the Galaxy *sells rather better* than the Encyclopedia Galactica."

If the explosion of new iPhone apps over the last year is any indication, you will want to take advantage of the iPad's sexiness, and that means leveraging its fabulous touch-sensitive interface and other features. Because the iPad evolved from the iPhone design, the iPad has design advantages that make netbooks and laptops feel like the dull Encyclopedia Galactica. Most iPhone apps are designed to take advantage of the iPhone's Multi-Touch display; accelerometer (which detects acceleration, rotation, motion gestures, and tilt); or location services for detecting its physical location — or all three.

But you can create iPad apps that are not just a little bit better than their iPhone counterparts, but a *lot* better (and an order of magnitude more powerful), with an interface that's simpler to use than a Mac.

Providing an immersive experience

An iPad app can offer a more immersive experience compared with an iPhone app by adding *more content* — full pages from the Internet or in memory, maps you can zoom into, full-screen videos and slideshows with music, and so on. People can enjoy this content while away from their desks — on living room couches, in coffee shops, on the train, in outer space — and more easily share it with others, far more easily than they can with an iPhone or iPod touch.

Whenever possible, add a realistic, physical dimension to your application. *The New York Times,* for example, designed an iPad app (refer to Figure 1-1, right side) that offers an immersive experience with the newspaper that includes truly embedded, fully functional videos (not just videos that appear in a separate window), and lets you tap the page to change the layout of columns, resize the text with a pinch, or show pop-up menus for more stories.

Electronic Arts created a version of its popular game Need for Speed for the iPad that feels like you're driving the display with your hands as you steer the car using the iPad like a steering wheel. The high-definition screen is just inches from your face — the field of view and the sensation of speed you get is incredible. The full-screen display is also fully touch sensitive — you can tap on a car and see inside it, flick a lifelike gear shifter to shift gears, and tap the rear-view mirror to look behind you.

Even utility apps can be rethought to be a better experience. On the iPhone, the Contacts app is a streamlined list, but on the iPad, Contacts is an address book with a beautifully tangible look and feel. The more true to life your application looks and behaves, the easier it is for people to understand how it works and the more they enjoy using it.

Making content relevant

An iPad app can present information relevant to where you are, what time it is, what your next activity might be, and how you're holding the device (in portrait or landscape view, tilting and shaking it, and so on), just like an iPhone or iPod touch app.

For example, the version of Google Maps for the iPad displays a full-screen map that can show your location and immediately find commercial establishments nearby. (For example, you can search for "sushi" to find sushi restaurants.)

The iPad platform offers a strong foundation for pinpointing the device's current location on a map, controlling views, managing data, playing multimedia content, switching display orientations, and processing gestures. Because the iPad platform can do all that, an app can know your current location, the hotels or campgrounds you're going to stay at, and the planets you're planning to visit. It can even show videos and play the music of the stars all at the same time. Rather than orbiting some moon while searching maps and brochures, you can know at a glance where you are, how to get to your destination, and what the weather's like so that you know what to wear.

Designing for the touch-display experience

The important design decision to make, whether you're starting from scratch with a new iPad app or evolving one from an iPhone app, is to use the large iPad screen and the new user interface elements to give people access to more information in one place. Although you don't want to pack too much information into one screen, you also want to prevent people from feeling that they must visit many different screens to find what they want. An iPad app can offer the primary content on the Main view and provide additional information or tools in an auxiliary view (such as a *popover* that appears semi-transparently above the Main view) to give users access to functions without requiring them to leave the context of the Main view.

The large iPad screen also gives you a lot more room for multifinger gestures, including gestures made by more than one person. An iPad app can react to gestures and offer touch controls and pop-up settings that are relevant to what you're actually doing in the app and where you place your fingers. With a display the size of a netbook, you have a lot more screen real estate to allow dragging and two-finger gestures with graphics and images, and depending on what you're doing, a tap or gesture on a particular part of the screen can have a particular function. For example, in the Gameloft version of the first-person shooter called *Nova* (as adapted to the iPad), the display size gives you more flexibility than the iPhone version, with more controls and objects such as mini-maps, and you can slide two fingers across the screen to throw grenades.

With all this in mind, there are at least two things that you need to consider — besides functionality, of course — when it comes to creating a great iPad app:

- Exploiting the platform and ecosystem
- Creating a compelling user experience

The rest of this chapter and Chapter 2 dig more into this Two-Part Rule of Great iPad Applications.

Exploiting the Platform

Okay, enough talk about the iPad's unique experience. Just what exactly is the iPad platform, and what are its features?

The iPad runs iPhone OS 3.2 as its operating system, and iPad apps use many of the same views and controls you used if you already developed an iPhone app. But the design similarities end there. The iPad's *hardware* is ground zero for conceiving the design of an iPad app — it's the place to start dreaming of what kind of experience to provide:

- A touch-sensitive display size of 1,024 x 768 pixels that supports multifinger gestures.
- The connection features of the iPhone (except phone calls): Wi-Fi and optional 3G Internet access; a compass; location services (although a hardware GPS isn't included in the first version of the iPad, so it isn't as accurate); and the ability to play audio and video with ease.

- Flexible orientation — users can tilt it, rotate it, and turn it upside down.
- The capability to plug in an external keyboard and use it in place of the onscreen keyboard for extended typing.
- The ability for users to dock the iPad and share files with a computer or other iPad users.

Exploiting advantages of the system

One of the keys to creating a great app is taking advantage of what the device offers. In the case of a new platform with new possibilities, such as the iPad, exploiting advantages is especially important. The combination of hardware and system software open up design advantages that depart from the typical design approach for desktop and laptop applications.

For example:

- **Multifinger gestures:** Applications respond to multifinger gestures, not mouse clicks. If you design an app that simply uses a single finger tap as if it were a mouse click, you may be missing an opportunity to design a better user experience.
- **Movement and orientation:** The iPad includes an accelerometer just like an iPhone and iPod touch, so you can also design apps that detect accelerated movement, as well as change the display for different orientations.
- **Split views and unique keyboards:** You can use a Split view to display more than one view onscreen at a time. You can also bring up a special keyboard unique to the task, such as the numbers-and-formulas keyboard that appears in the Numbers app for the iPad.
- **Internet access:** Users can send and receive e-mail and browse the Web, as well as sync contacts, calendars, and notes over the Internet, and download content from Apple stores, just like an iPhone or iPod touch.
- **Computer sync over USB connection or local area network:** Users can sync their photos, contacts, calendars, music, video, and other content from their computers (again, just like an iPhone or iPod touch), and with some apps (such as Bento from FileMaker), you can sync data over a local area network.
- **Television or projection system connection:** Users can connect the iPad to an HDTV or projection system in order to show content to larger audiences.

- **Consistent system environment:** The Home button quits your app, and the volume controls take care of audio, just like you'd expect them to. User preference settings can be made available in the Settings application (to avoid cluttering up your app's user interface) and your iPad and iPhone/iPod touch apps can coexist on an iPad with Web services and apps created in HTML5.
- **Breathtaking imagery:** Photos and video already look fantastic on this display, but the artwork you create yourself for your app should be set to 24 bits (8 bits each for red, green, and blue), plus an 8-bit alpha channel to specify how a pixel's color should be merged with another pixel when the two are overlaid one on top of the other. In general, the PNG format is recommended for graphics and artwork.

In the following sections, you get to dive into some of the major features, grouped into the following major areas:

- Accessing the Internet
- Tracking location
- Tracking motion
- Supporting multifinger gestures and touches,
- Playing content
- Accessing the content of Apple's supplied apps (such as Contacts and Photos)
- Taking advantage of the iPad display.

Accessing the Internet

An iPad can access Web sites and servers on the Internet through Wi-Fi or optional 3G services. This Internet access gives you the ability to create apps that can provide real-time information. An app can tell a user, for example, that the next tour at the Tate Modern in London is at 3 p.m.

This kind of access also allows you, as the developer, to go beyond the limited memory and processing power of the device and access large amounts of data stored on servers, or even offload the processing. You don't need all the information for every city in the world stored on your iPad. You can send the request to a server for all that information, especially information that changes often.

This technique is called *client-server computing* — a well-established software architecture where the client provides a way to make requests to a server on a network that's just waiting for the opportunity to do something. A Web browser is an example of a client accessing information from other Web sites that act as servers.

Knowing the location of the user

You can create an app that can determine the device's current location or even be notified when that location changes, using the iPad's location services. As people move, it may make sense for your app to tailor itself to where the user is, moment by moment.

Many iPad and iPhone apps use location information to tell you where the nearest coffee house is or even where your friends are. The iPadTravel411 sample application described in Part V uses this information to tell you where *you* are and give you directions to your hotel.

When you know the user's location, you can even put it on a map, along with other places he or she may be interested in. You find out how easy it is to add a map to your app in Chapter 14.

Tracking orientation and motion

The iPad contains three *accelerometers* — devices that detect changes in movement. Each device measures change along one of the primary axes in three-dimensional space. An app can, for example, know when the user has turned the device from vertical to horizontal, and it can change the view from portrait to landscape if doing so makes for a better user experience.

You can also determine other types of motion such as a sudden start or stop in movement (think of a car accident or fall) or the user shaking the device back and forth. It makes some way-cool features easy to implement — for example, the Etch-A-Sketch metaphor of shaking the iPad to undo an operation. You can even control a game by moving the iPad like a controller — such as the aforementioned Need for Speed game for the iPad (Electronic Arts), in which you drive the car by using the iPad like a steering wheel.

Tracking user's fingers on the screen

People use their fingers to select and manipulate objects on the iPad screen. The moves that do the work, called *gestures,* give the user a heightened sense of control and intimacy with the device. Several standard gestures — tap, double-tap, pinch-close, pinch-open, flick, and drag — are used in the applications supplied with the iPad.

TIP

You may want to stick with the standard gestures in your app, just because folks are already aware of (and comfortable with) the current pool, but the iPad's multifinger gesture support lets you go beyond standard gestures when appropriate. Because you can monitor the movement of each finger to detect gestures, you can create your own.

Playing content

Your iPad app can easily play audio and video. You can play sound effects or take advantage of the multichannel audio and mixing capabilities available to you. You can also play back many standard movie file formats, configure the aspect ratio, and specify whether controls are displayed. You can put up pages that look like Web pages or book pages if you want, and you can easily mix content for an immersive experience.

Accessing information from Apple's apps

Your app can access the user's information in the Contacts app on the iPad and display that information in a different way or use it as information in your application.

Your app can also access the Photo library in the iPad Photos app, not only to display them, but also to use (or even modify) them. For example, the Photos app lets you add a photo to a contact, and several applications enable you to edit your photos on the iPad itself.

Living large on the big screen

The iPad display offers enough space to show a laptop application (which is one reason why Web pages look so great). You can organize your app with a master and detailed list, or a source list layout (with a view) similar to the Mac OS X versions of iTunes and iPhoto and exemplified by the Photos, Contacts, and Keynote apps on the iPad.

If you're familiar with iPhone apps and Mac OS X applications, think somewhere in-between. With the iPad touch-sensitive display, you no longer have to create different screens of menus (as you might for an iPhone app) or deploy drop-down menus and toolbars (as you might for an Mac OS X app) to offer many functions.

For example, to crop and mask out parts of an image in Apple's Keynote app for the iPad (which lets you create slideshows), you don't have to select a photo and then hunt for the cropping tool or select a menu item — just double-tap the image, and a mask slider appears. In Apple's Numbers app for the iPad, if you double-tap a numeric formula, the app displays a special numeric and function keyboard rather than a full text keyboard — and the app can recognize what you're doing and finish the function (such as a Sum function) for you.

These are examples of redesigning a known type of application to get rid of (or at least minimize) that modal experience of using a smartphone app — in which you have only one path of communication to perform a task or supply a response. iPad applications should allow people to interact with them in nonlinear ways. Modality prevents this freedom by interrupting a user's workflow and forcing the user to choose a particular path.

Lists are a common way to efficiently display large amounts of information in iPhone apps. Lists are very useful in iPad apps, too, but you should take this opportunity to investigate whether you can present the same information in a richer way on the larger display.

Embracing the iPad's Limitations

Along with all those features, however, the iPad has some limitations. The key to successful app development — and to not making yourself too crazy — is to understand those limitations, live and program within them, and even learn to love them. (It can be done. Honest.) These constraints help you understand the kinds of applications that are right for this device.

REMEMBER

Often, it's likely that if you *can't* do something (easily, anyway) because of the iPad's limitations, then maybe you shouldn't.

The iPad evolved from the iPhone and iPod touch, and there are related limitations you need to consider, as well as a few things left out. So learn to live with and embrace some facts of iPad life:

- Users have fat fingers. That may be easy to deal with, but keep in mind that you may want to design a multiuser app for the iPad that takes into account multiple fingers. (Anyone for a nice game of air hockey?)
- Memory and battery power are limited, just like an iPhone or iPod touch. This limitation may or may not be a decisive factor, depending on what kind of app you want to create, but smaller apps generally perform better.
- Users can run only one application at a time — again, just like an iPhone or iPod touch. Although this limitation may change in the future, you need to keep this in mind before designing an app that relies on another app (such as one that offers links that can open the Safari browser).
- A camera isn't included in the first version of the iPad, but your iPad app can access the synced Photo library as well as synced contacts.

The next sections help get you closer to a state of iPad enlightenment.

Designing for fingers

Although the Multi-Touch interface is a feature of both the iPad and the iPhone/iPod touch, it brings with it some limitations — although not as many as with the smaller iPhone/iPod touch displays.

First of all, fingers aren't as precise as a mouse pointer, which makes some operations even more difficult on an iPhone or iPod touch than on an iPad (text selection, for example). Still, due to fat fingers, user-interface elements need to be large enough and spaced far enough apart so that users' fingers can find their way around the interface comfortably. Apple recommends that anything a user has to select or manipulate with a finger be a minimum of 44 x 44 pixels in size.

Because it's so much easier to make a mistake using fingers, you also need to ensure that you implement a robust — yet unobtrusive — Undo mechanism. You don't want to have your users confirm every action (it makes using the app tedious), but on the other hand, you don't want your app to let anybody mistakenly delete a page without asking, "Are you *sure* this is what you *really* want to do?" Lost work is worse than tedious. Fortunately, the iPad supports the same shake-to-undo feature as the iPhone.

Balancing memory and battery life

As an app designer for the iPad, you have several balancing acts to keep in mind:

- Although significant by the original Macintosh's standards, the computer power and amount of memory on the iPad are limited.
- Although access to the Internet can mitigate the power and memory limitations by storing data and (sometimes) offloading processing to a server, those Internet operations eat up the battery faster.
- Although the iPad power-management system conserves power by shutting down any hardware features that are not currently being used, a developer must manage the trade-off between all those busy features and shorter battery life. Any app that takes advantage of Internet access, core location, and a couple of accelerometers is going to eat up the batteries.

The iPad is particularly unforgiving when it comes to memory usage. If you run out of memory, in order to prevent corruption of other apps and memory the system will simply *shut down your app* (unfortunately not to the tune of "Shut Down" by the Beach Boys).

This just goes to show that not *all* limitations can be exploited as "features."

Why Develop iPad Applications?

Because you can. Because it's fun. And because the time has come (today!). Developing iPad apps can be the most fun you've had in years, with very little investment of time and money (compared with developing for platforms like Windows). Here's why:

- **iPad apps are usually bite-sized, which means they're small enough to get your head around.** A single developer — or one with a partner and maybe some graphics support — can do them. You don't need a 20-person project with endless procedures and processes and meetings to create something valuable.
- **The apps use the most innovative platform available for mobile computing.** The iPad is a game-changer. It's completely changing the Internet as a publishing medium, the software industry with regard to applications, and the mobile device industry with regard to the overall digital media experience.
- **The free Software Development Kit (SDK) makes development as easy as possible.** This book reveals the SDK in all its splendor and glory. If you can't stand waiting, you *could* go on to Chapter 4, register as an iPhone/iPad developer, and download the SDK . . . but (fair warning) jumping the gun leads to extra hassle. It's worth getting a handle on the ins and outs of iPad app development beforehand.

The iPad has three other advantages that are important to you as a developer:

- **You can distribute your app through the App Store.** Apple will list your app in the App Store in the category you specify, and the store takes care of credit-card processing (if you charge for your app), hosting, downloading, notifying users of updates, and all those things that most developers hate doing. Developers name their own prices for their creations or distribute them free; Apple gets 30 percent of the sales price of commercial apps, with the developer getting the rest. However, keep in mind that Apple must approve your app before it appears in the App Store — see Chapter 6 for details on submitting your app and jumping through the hoops to get it approved.
- **Apple has a developer program.** To get your app into the store, you have to pay $99 to join the iPhone Developer Program (which includes iPad development support). (An enterprise pays $299 to join up.) But that's it. There are none of the infamous hidden charges that you often encounter, especially when dealing with credit-card companies. Go to the Apple iPhone Developer site (`http://developer.apple.com/iphone/program`) and click the Enroll Now button. Chapter 6 describes how to work with the App Store to get your apps published.

- **It's a business tool.** The iPad has become an acceptable business tool, in part because it has tight security as well as support for Microsoft Exchange and Office. This happy state of affairs expands the possible audience for your application.

Developing with Apple's Expectations in Mind

Just as the iPad can extend the reach of the user, the device possibilities and the development environment can extend your reach as a developer. It helps to understand Apple's perspective on what iPad apps should be — the company clearly has done some serious thinking about it, far longer than anybody else out there, having taken years to bring the iPad to market under a veil of secrecy.

So what does Apple think? Spokespeople often talk about three different application styles:

- **Productivity applications use and manipulate information.** The iPadTravel411 sample app that I show in this book is an example, and so are Bento (FileMaker), and Apple's iWork apps — Keynote, Pages, and Numbers. Common to all these apps is the use and manipulation of multiple types of information. (I'm not talking about the Productivity category in the App Store — that's a marketing designation.)
- **Utility applications perform simple, highly defined tasks.** The preinstalled Weather app is an example — it deals only with the weather data. The Brushes app for painting (Steve Sprang) is considered a utility, as it performs a simple, highly defined task. (Again, I'm not talking about the Utilities category in the App Store, although many of those apps are considered utility apps because they perform simple, highly defined tasks.)
- **Immersive applications are focused on delivering — and having the user interact with — content in a visually rich environment.** A game is a typical example of an immersive application.

Although these categories help you understand how Apple thinks about iPad apps (at least publicly), don't let them get in the way of your creativity. You've probably heard *ad nauseam* about stepping outside the box. But hold on to your lunch; the iPad "box" isn't even a box yet. So here's a more extreme metaphor: Try diving into the abyss and coming up with something really new.

An Overview of the Development Cycle

To keep from drowning in that abyss, you need a plan to guide you through it. Socrates anticipated software development when he said that there's nothing stable in human affairs. Tacitus, with more data in hand 450 years later, saw that in all things there is a law of cycles. By the late 1960s, the Jefferson Airplane singers were singing, "go with the natural flow, like water off a spinning ball."

In plain words, your software development plan is a cycle; perhaps a vicious cycle, but it can be a cycle through the park. You may repeat procedures within the cycle iteratively until you get it right, but the key to understanding the cycle is the recognition that once you spin off version 1 of your app, you start all over again to develop an update.

In general terms, the software development cycle is the process of creating or altering a software product or service. Theorists have created models and methodologies for defining this cycle. Although there are at least half a dozen models (Neal's a recovering software development methodologist), the one I go through here is pretty simple and is well suited for the iPad to boot. Here goes:

1. Defining the problems
2. Designing the user experience
    a. Understanding the real-world context
    b. Understanding the device context
    c. Categorizing the problems and defining the solutions
3. Creating the program architecture
    a. A Main view
    b. Content views
    c. View controllers
    d. Models
4. Writing the code
5. Doing it until you get it right

Of course, the actual analysis, design, and programming (not to mention testing) process has a bit more to it than this — and the specification and design definitely involve more than what you see in this book. But from a process perspective, it's pretty close to the real thing. It does give you an idea of the questions you need to ask — and have answered — in order to develop an effective iPad application.

A word of caution, though. Even though iPad apps are much easier to get your head around than, say, a full-blown enterprise service-oriented architecture, they come equipped with a unique set of challenges. Between the iPad platform limitations and the high expectation of all the new iPad users, you'll have your hands full.

The Sample Applications

It's hard enough to understand how to develop an app, and even harder if the first example you turn to is too complex to get your head around. The first sample app, DeepThoughts (shown in Figure 1-2), which you find out how to build in Part IV, is simple enough to understand, and yet it demonstrates enough of the building blocks for creating a sophisticated iPad app that you should have no trouble following along and building it. With a little more (although not much more) work, you can use the development environment to actually create something of value.

DeepThoughts displays whatever text you enter in a flowing animation that fills the display, supposedly suggesting a meditative state (as in "peace love groovy music"). You can speed up or slow down the animation by swiping left or right, so you find out how to deal with that simple gesture, as well as with tapping an Info button or the display to change the text and speed.

After you know a bit more about the application design cycle and what makes a good user interface, and even more (actually quite a bit more) about the iPad technologies that work behind the screen — such as frameworks, windows, views, and view controllers — and then just a few more details about getting your app ready for the App Store and the public, you're ready to do some real coding — the DeepThoughts app.

After that, you find out about the design of the iPadTravel411 app (shown in Figure 1-3), starting in Chapter 13. You find out how to use a Split view, present a map, work with Table views, add content, access data on the Web, include data with your app, and allow users to set preferences.

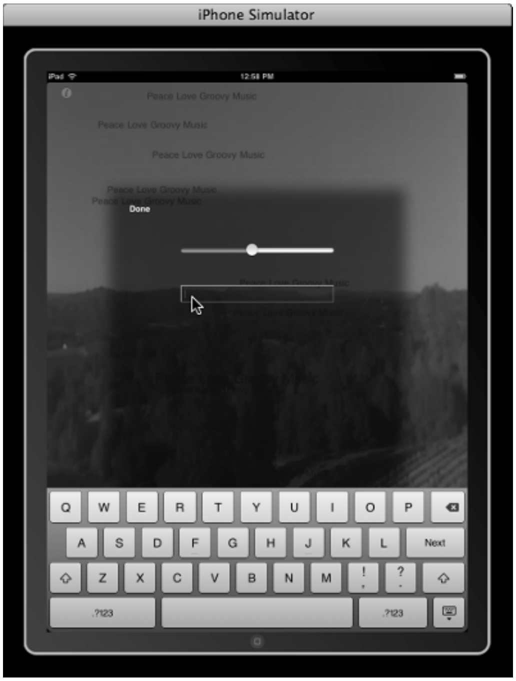

Figure 1-2: This book will provoke Deep Thoughts.

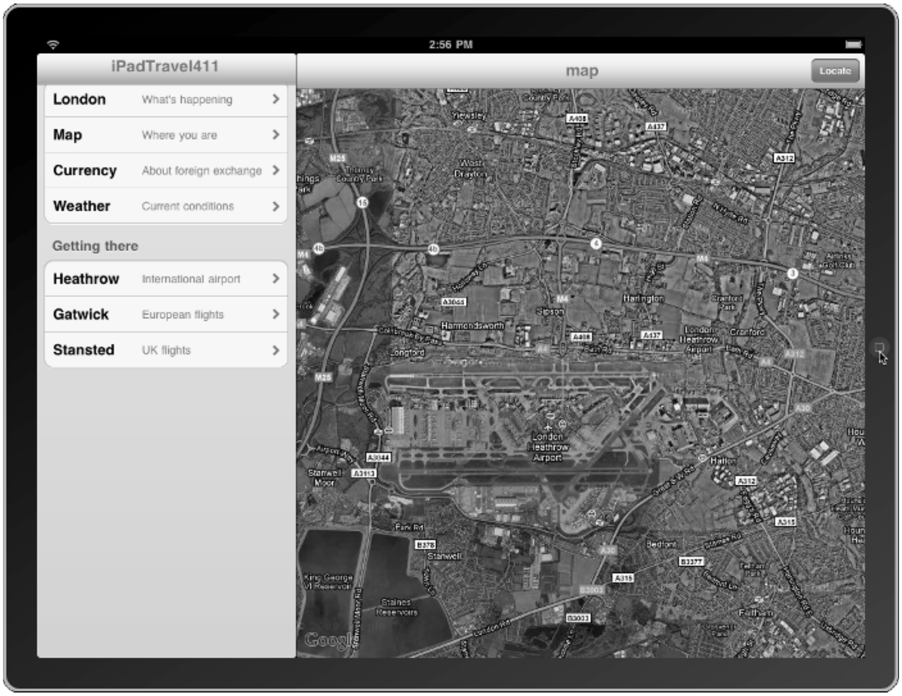

Figure 1-3: This sample app may provoke a trip to London.

What's Next

You must be raring to go now and just can't wait to download the Software Development Kit (SDK). That's exactly what many new developers do — and later are sorry that they didn't spend more time upfront understanding the iPad user experience, how applications work in the iPad environment, and the guidelines that Apple enforces for apps to be approved for the App Store.

So be patient. The Hitchhiker's Guide to the Galaxy, that wonderful fantasy of an iPad from 1979, suggests that since space is "big, really big . . . you just won't believe how vastly hugely mind-bogglingly big it is," and suggests that you bring a towel. The guide says a towel "is about the most massively useful thing an interstellar hitchhiker can have." (Again with Douglas Adams? But I promise not to get into the meaning of life, the universe, and everything, or the ultimate question — for which the answer is 42.) This book is your towel for the journey. The following chapters cover all the aspects of development you need to know before you spend time coding. Then, I promise, it's off to the stars.