Chapter 5

Autorotation and Autosizing

The iPhone, iPad, and other iOS devices are all amazing pieces of engineering. Apple engineers found all kinds of ways to squeeze maximum functionality into a pretty dam small package. One example is the mechanism that allows applications to be used in either portrait (tall and skinny) or landscape (short and wide) mode, and to change that orientation at runtime if the phone is rotated. A prime example of this behavior, which is called autorotation, can be seen in iOS’s web browser, Mobile Safari (see Figure 5–1).

Figure 5–1. Like many iOS applications, Mobile Safari changes its display based on how it is held, making the most of the available screen space.
In this chapter, we’ll cover autorotation in detail. We’ll start with an overview of the ins and outs of autorotation.

**The Mechanics of Autorotation**

Autorotation might not be right for every application. Several of Apple’s iPhone applications support only a single orientation. Movies can be watched only in landscape mode, for example, and contacts can be edited only in portrait mode. However, that’s not true for the iPad, for which Apple recommends that essentially all applications (with the possible exception of games that are designed around a particular layout) should support every orientation.

In fact, all of Apple’s own iPad apps work fine in both orientations. Many of them use the different orientations to show you different views of your data. For example, the Mail and Notes apps use landscape orientation to show a list of items (folders, messages, or notes) on the left and the selected item on the right, and portrait orientation to let you focus on the details of just the selected item.

The bottom line is that if autorotation enhances the user experience, add it to your application. Fortunately, Apple did a great job of hiding the complexities of autorotation in iOS and in the UIKit, so implementing this behavior in your own iOS applications is actually quite easy.

Autorotation is specified in the view controller, so if the user rotates the device, the active view controller will be asked if it’s OK to rotate to the new orientation (which you’ll see how to do in this chapter). If the view controller responds in the affirmative, the application’s window and views will be rotated, and the window and view will be resized to fit the new orientation.

On the iPhone and iPod touch, a view that starts in portrait mode will be 320 pixels wide and 480 pixels tall. On the iPad, portrait mode means 768 pixels wide and 1024 pixels tall. The amount of screen real estate available for your app will be decreased by 20 pixels vertically if your app is showing the status bar. The status bar is the 20-pixel strip at the top of the screen (see Figure 5–1) that shows things like signal strength, time, and battery charge.

When the phone is switched to landscape mode, the view rotates, along with the application’s window, and is resized to fit the new orientation, so that it is 480 pixels wide by 320 pixels tall (iPhone and iPod touch) or 1024 pixels wide by 768 pixels tall (iPad). As before, the vertical space actually available to your app is reduced by 20 pixels if you’re showing the status bar, which most apps do.