Capitolo 2

Ciao iPhone

iPhone è una piattaforma sorprendente, ricca di tecnologia avanzata che fa venir voglia di creare qualcosa. L'App Store è pieno di applicazioni che dovrebbero ispirare e porsi come sfide per gli sviluppatori. Le possibilità sono innumerevoli.

Cominciamo con una semplice applicazione del tipo "Ciao a tutti". Nel resto del libro esamineremo molte delle tecnologie e delle API disponibili. Vedrete tutte le funzionalità, dal rilevamento della posizione all'accelerometro, dai controller alle animazioni, ma in questo capitolo ci limitiamo agli strumenti e al ciclo di sviluppo di base per creare applicazioni destinate a iPhone.

La nostra prima applicazione è volutamente semplicissima, si limita a visualizzare il testo statico: "Ciao iPhone" sullo schermo. Per prima cosa verificherete di avere installato tutti gli strumenti da sviluppatore necessari e poi ne utilizzerete due per creare ed eseguire l'applicazione nel simulatore di iPhone. Poiché quest'ultimo è un'applicazione iPhone, farete ruotare il testo quando il dispositivo viene ruotato di lato. Infine, scaricherete questa applicazione sul vostro iPhone e la eseguirete.

In questo capitolo

2.1 Per cominciare

2.2 Creazione del progetto Ciao iPhone

2.3 Creazione di una semplice interfaccia

2.4 Rotazione del testo: modifiche all'interfaccia utente

2.5 Rotazione del testo: modifica del codice

2.6 Navigazione nel codice

2.7 Esecuzione dell'applicazione su iPhone

2.1 Per cominciare

Se avete già sviluppato applicazioni per Mac OS X, avete familiarità con Xcode, Interface Builder e gli altri strumenti. Sviluppare applicazioni per iPhone è diverso perché il dispositivo di destinazione è differente, ma molti degli strumenti, delle API e delle competenze richieste coincidono. Potrete leggere questo capitolo tutto d'un fiato.

Terminologia

Da qui in avanti quando parleremo di iPhone intenderemo qualsiasi dispositivo con iPhone OS; perciò, anche chi utilizza un iPod touch non deve sentirsi escluso.

Se vi avvicinate a iPhone da un'altra piattaforma, siete i benvenuti. Avete certamente utilizzato un IDE (Integrated Development Environment): per gli sviluppatori iPhone, tale ruolo è svolto da Xcode, che fornisce tutte le funzionalità attese quali gestione dei file, compilazione, debugging e rapporti di errore. Quando conoscerete Xcode lo troverete familiare come l'IDE a cui siete già abituati. Se invece siete del tutto nuovi al mondo dello sviluppo, credo che lo troverete divertente e appagante.

Ci sono tre strumenti principali che utilizzerete sempre durante lo sviluppo per iPhone: Xcode, Interface Builder e Instruments. Utilizzerete Xcode per gestire il vostro progetto e scrivere, eseguire ed effettuare il debugging del codice. Utilizzerete Interface Builder per creare l'interfaccia e collegarla al codice, mentre Instruments vi aiuterà a individuare le aree critiche per quanto riguarda le prestazioni.

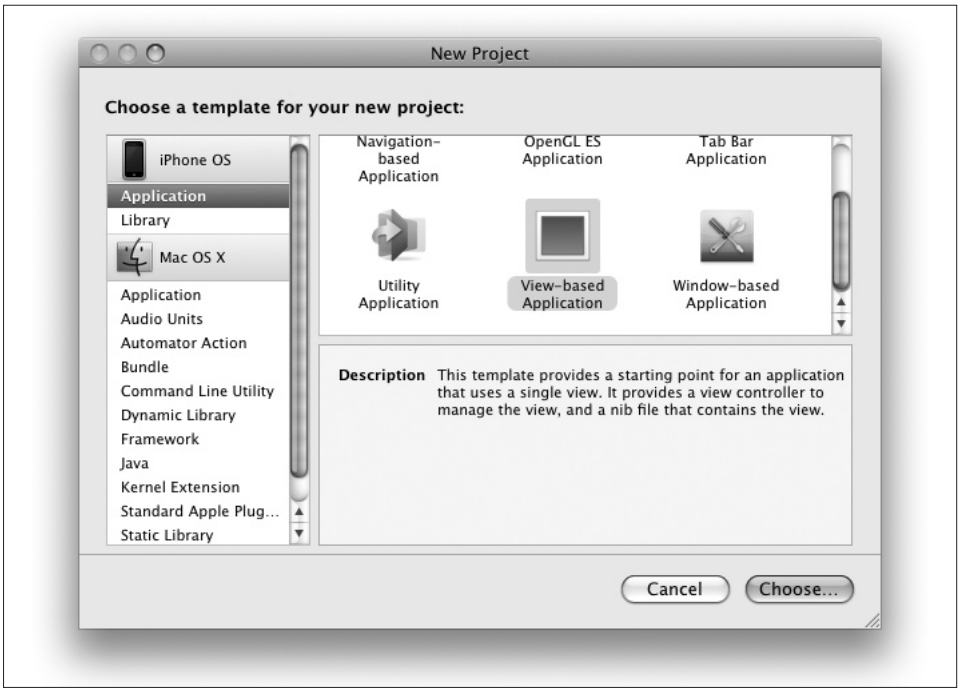
In questo libro non esamineremo tutti i dettagli di questi strumenti, ma vi mostreremo come utilizzarli e metteremo in evidenza le funzionalità richieste per sviluppare applicazioni che ottengano il loro scopo. Nel caso in cui vogliate approfondire una particolare caratteristica di Xcode, potete consultare l'eccellente documentazione integrata di Apple.

Assicuratevi di aver installato gli strumenti di sviluppo. Se non lo avete già fatto, prelevate l'iPhone SDK da `http://developer.apple.com/iphone`. Dovrete registrarvi come sviluppatore iPhone, ma la registrazione è gratuita, come l'SDK. È richiesta la cifra di 99 dollari per partecipare all'iPhone Developer Program, ma è necessario farlo per ottenere un certificato che consente di eseguire le applicazioni sul dispositivo. Dopo aver prelevato il pacchetto, installatelo facendo doppio clic sul file *.dmg* e poi sul pacchetto *iPhone SDK*. Seguite le istruzioni fornite sullo schermo e scegliete la posizione predefinita per installare gli strumenti per sviluppatori. Una volta completata l'installazione troverete Xcode in */Developer/Applications/Xcode.app*. Se scegliete un'altra directory root per l'installazione degli strumenti per sviluppatori, naturalmente il percorso varierà di conseguenza.

Avviate Xcode facendo doppio clic su di esso. È comodo mantenere Xcode, Interface Builder e Instruments nel dock per un facile accesso. Quando avviate Xcode vedete la finestra di benvenuto, che offre molte informazioni. In alto vedete diverse sezioni, ognuna delle quali porta a una serie di punti da cui raggiungere documentazione o altre informazioni utili. Personalmente utilizzo la sezione 'Xcode News [RSS]' per tenermi informato sui nuovi esempi e documenti forniti da Apple. La sezione 'iPhone Dev Center' è ricca di collegamenti alle varie sezioni del centro di sviluppo online. Mantenete questa finestra attiva all'avvio almeno finché non avrete acquisito una discreta familiarità con Xcode. Spendete un po' di tempo per esaminare ciascuna delle sezioni e osservarne il contenuto, è il modo migliore per apprendere Xcode.

2.2 Creazione del progetto Ciao iPhone

Ora che Xcode è in esecuzione, create il vostro primo progetto. Selezionate *File > New Project* o premete Mela+Maiusc+N per avviare la procedura guidata, quindi scegliete *iPhone OS > Application* per vedere l'elenco dei template (modelli) di progetti iPhone.



Esamineremo molti dei template disponibili nei vari esempi presentati nel libro; per ora scegliete il template *View-Based Application* e fate clic sul pulsante *Choose*, quindi nel pannello che si apre scegliete una cartella e assegnate al progetto il nome 'Hello'. Xcode utilizza il template *View-Based Application* per creare un'applicazione con una singola view. Vengono creati automaticamente diversi file, ma ne esamineremo molti in seguito, perciò ora non è il caso di occuparsene. La finestra dovrebbe essere simile a quella della Figura 2.1. Sul lato sinistro vi è un elenco di gruppi organizzativi; ognuno di questi contenitori può essere utilizzato per inserire degli elementi in modo da facilitare l'organizzazione. I gruppi non sono implicitamente legati a una struttura di file system, perciò vi è un'ampia flessibilità per organizzare codice e risorse nel modo più opportuno. Per chi muove i primi passi spesso risulta più semplice apprendere la configurazione iniziale, ma nulla vieta di utilizzarne una diversa.

Lungo la parte superiore della finestra si trova la barra degli strumenti, uno spazio configurabile dove potete inserire i comandi che utilizzate con maggiore frequenza in modo che risultino facilmente accessibili. Nel mio caso utilizzo la casella a discesa *Overview* (panoramica) che mi consente di passare rapidamente tra il simulatore di iPhone e il dispositivo vero e proprio come destinazione in cui installare la mia applicazione; se non vedete una casella di riepilogo *Overview* vi consigliamo di aggiungerla (alcuni sviluppatori la vedono già per default, naturalmente in questo caso non occorre fare nulla).

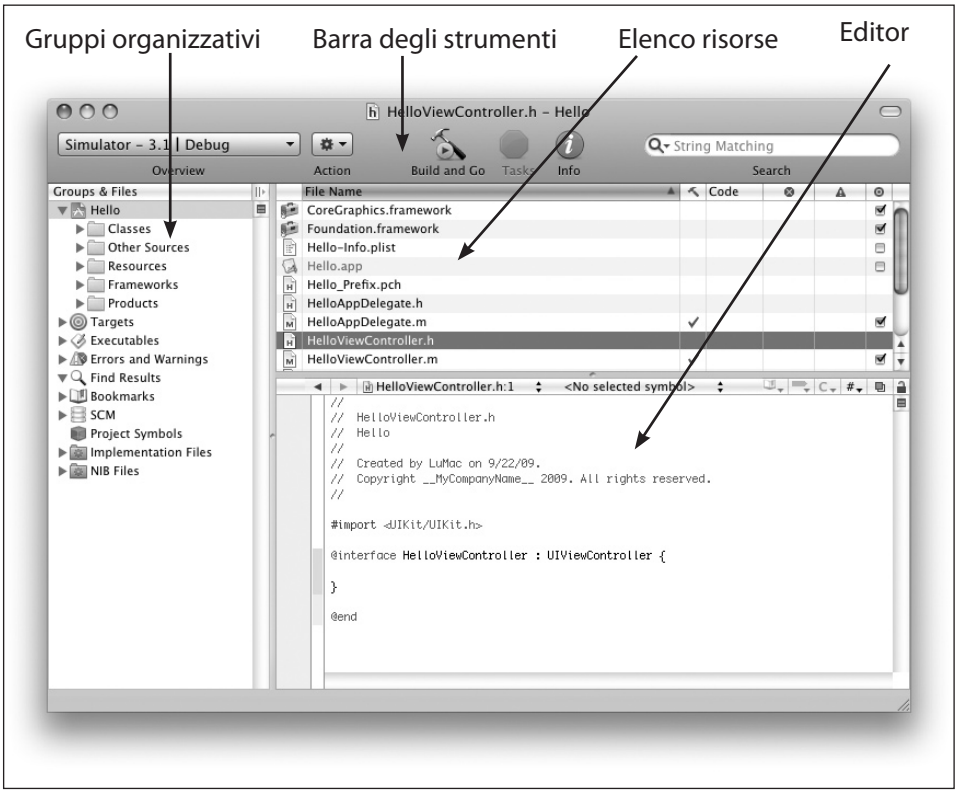


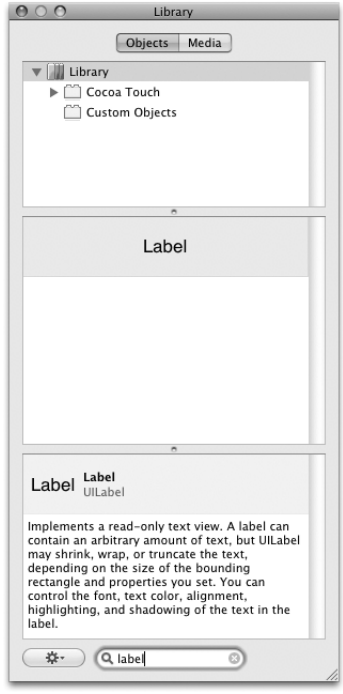
Figura 2.1 Il progetto Ciao iPhone.

Per personalizzare la barra degli strumenti selezionate *View > Customize Toolbar* o premete il tasto Ctrl e fate clic sullo sfondo della barra, quindi scegliete *Customize Toolbar*. Selezionate l'elemento *Overview* e trascinatelo sulla barra: il cursore assumerà la forma di un cerchio verde con il segno '+' all'interno. Rilasciate l'elemento sul lato sinistro della barra.
Sul lato destro, sotto la barra degli strumenti si trova l'elenco di file/risorse, che mostra i file che fanno parte del gruppo selezionato nell'elenco *Groups & Files* a sinistra. Poiché il progetto è attualmente selezionato, sono elencati tutti i file corrispondenti. Quando si fa clic su un file di questo elenco, il suo nome appare appena sotto il pannello dell'editor pane. Questo elenco segue la selezione effettuata in *Groups & Files*, quando si cambia il gruppo selezionato, vengono visualizzati solo i file che ne fanno parte. Il pannello dell'editor è quello utilizzato per gran parte del lavoro di stesura del codice e consente di accedere alla maggior parte delle funzioni legate al codice (completamento, refactoring e così via).
Una volta completato il giro panoramico di Xcode, eseguiamo la nostra nuova applicazione per vedere come si comporta. Fate clic sul pulsante *Build and Go* (o premete Mela+Invio) e Xcode si occuperà del resto. Le applicazioni iPhone, come quelle per Mac, sono costituite da speciali directory chiamate *app bundle*, o *bundle di applicazione*. Durante il processo di compilazione, Xcode crea un bundle di applicazione con il vostro codice compilato e le risorse copiate nelle posizioni appropriate.
Dopo che l'applicazione è stata compilata e installata nel simulatore da XCode, viene avviata nel simulatore stesso. Nella prima esecuzione si vedrà semplicemente uno sfondo

grigio chiaro: non è molto, ma è quello che ci aspettavamo dato che non abbiamo ancora personalizzato il template. Ora apportiamo qualche modifica all'interfaccia.

2.3 Creazione di una semplice interfaccia

È giunto il momento di conoscere Interface Builder, uno dei principali strumenti utilizzati per creare applicazioni iPhone e certamente il più importante per quanto riguarda il layout e la configurazione dell'interfaccia utente (UI, User Interface).
Per ora l'applicazione che stiamo considerando non deve fare nulla di concreto, perciò non abbiamo la necessità di scrivere codice; utilizzeremo strumenti visuali per creare l'interfaccia grafica. Interface Builder (IB), come si capisce dal nome, ci consente di creare le interfacce con cui i nostri utenti interagiranno. Lo strumento supporta gli oggetti UIKit utilizzati per realizzare interfacce e le linee guida HIG (Human Interface Guidelines) che ci aiutano a mantenere una buona coerenza tra le varie interfacce. È molto potente, per chi lo sa utilizzare. Come per Xcode, in questo libro non è possibile esaminare tutti i dettagli di IB, ma attraverso l'esperienza ottenuta realizzando gli esempi presentati qui, ne saprete abbastanza per poter approfondire voi stessi i dettagli che più vi interessano.
Ora costruiamo l'interfaccia per la nostra prima applicazione. Aprite *HelloView-Controller.xib* facendo doppio clic su di esso in Xcode (sotto il gruppo *Resources*). Vogliamo aggiungere un'etichetta all'interfaccia in cui inserire il nostro testo 'Ciao iPhone', perciò apriamo la libreria premendo Mela+Maiusc+L (o selezionando *Tools* > *Library*). La libreria raccoglie tutti gli oggetti dell'interfaccia e controller che potete istanziare nel file NIB. In questo caso cerchiamo un'etichetta, perciò digitiamo 'label' nel campo del filtro che si trova nella parte inferiore della finestra.



Trascinate l'etichetta nella view e spostatela fino ad agganciarla alle guide che si trovano sul lato sinistro.

CHE COSA SIGNIFICANO XIB E NIB?

Interface Builder supporta due tipi di file: *.xib* e *.nib*. L'estensione *.xib* indica semplicemente la versione XML del file *.nib*. Poiché il tipo di file .nib esiste già a partire da Mac OS X 10.0 (e anche prima, in NeXTStep), mentre i file *.xib* sono relativamente nuovi, anche questi spesso sono indicati semplicemente come file NIB, e noi in questo libro faremo lo stesso.

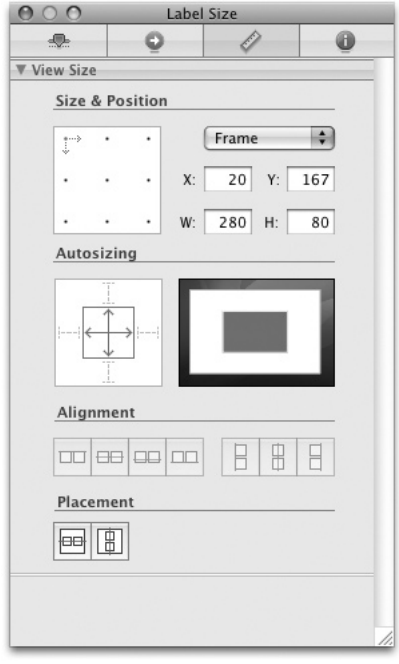
Selezionate l'etichetta, se non lo è già: i cerchietti blu attorno alla stessa consentono di ridimensionarla. Fate in modo che si estenda dalla guida di sinistra a quella destra, in modo che occupi l'intera larghezza della view meno il bordo determinato dalle guide. Modificate anche l'altezza portandola a 80 (potete farlo trascinando con il mouse, oppure aprendo la finestra di ispezione *Label Size* con Mela+3). Ora cambiate il tipo di carattere premendo Mela+T e scegliendo come dimensione 48. Fate doppio clic sull'etichetta per selezionare il testo e modificarlo, quindi digitate 'Ciao iPhone'. Potete anche modificare il colore del testo e aggiungere un'ombreggiatura e molti altri attributi utilizzando la finestra di ispezione *Attributes* (Mela+1). Inoltre, assicuratevi di indicare che volete centrare il testo, nella finestra degli attributi. Ecco come appare la finestra con il testo centrato, rosso (grigio in stampa) e ombreggiato:

Ora siete pronti a eseguire l'applicazione. Salvate le modifiche in Interface Builder, tornate a Xcode e fate clic sul pulsante *Build and Go* per controllare la vostra opera d'arte. All'avvio del simulatore dovreste vedere il testo dell'etichetta apparire al centro della view

su sfondo grigio. Nel simulatore selezionate da menu *Hardware* > *Rotate Left* o *Hardware* > *Rotate Right*: il dispositivo ruota a sinistra o a destra di novanta gradi e il testo ruota con esso. Apporteremo delle modifiche in modo che, quando l'utente ruota il dispositivo, il testo rimanga orientato in modo da essere sempre ben leggibile.

2.4 Rotazione del testo: modifiche all'interfaccia utente

Occorre eseguire due passaggi fondamentali per fare in modo che il programma risponda in modo appropriato quando il dispositivo viene ruotato. Dovrete impostare alcuni parametri sull'`UIView` in Interface Builder e modificare una sola riga di codice nella classe `HelloViewController` utilizzando Xcode. In questo paragrafo vedremo le modifiche all'interfaccia utente e nel prossimo apporteremo le modifiche al codice.
Selezionate l'etichetta e utilizzate la finestra di ispezione *Size* (Mela+3) per modificare la configurazione degli spazi e della dimensione della view quando si attua un ridimensionamento della view stessa o di quella che la contiene. Nel gergo di iPhone si parla di *strut* (letteralmente "puntone") e *spring* (letteralmente "molla"); il primo indica che la distanza tra la view e i bordi della view che la contiene (indicata anche come view contenitore o superview) rimane costante, mentre il secondo serve a specificare che la view può espandersi o contrarsi in orizzontale o in verticale. In questo esempio vogliamo mantenere l'etichetta centrata quando il dispositivo viene ruotato, perciò disattiviamo gli strut per tutti e quattro i bordi e attiviamo le spring per entrambe le direzioni. Quando il dispositivo ruota, cambiano le dimensioni della superview che contiene l'etichetta e perciò il sistema aggiornerà posizione e dimensioni di quest'ultima in base alle impostazioni effettuate. Dovete anche specificare l'allineamento del testo in modo che risulti centrato.
Nella figura seguente potete vedere la configurazione corretta per il nostro esempio.



Abbiamo terminato di aggiornare l'interfaccia utente, torniamo quindi a Xcode e apportiamo le modifiche necessarie al codice per fare in modo che l'applicazione risponda opportunamente alle rotazioni del dispositivo.

2.5 Rotazione del testo: modifica del codice

In Xcode, selezionate il gruppo *Classes* e poi nell'elenco dei file in alto a destra selezionate *HelloViewController.m*. Xcode apre il file nel pannello dell'editor sotto l'elenco. Trovate il metodo `shouldAutorotateToInterfaceOrientation:` (assicuratevi di rimuovere eventuali indicatori di commento), dove occorre modificare il codice per fare in modo che l'applicazione capisca che è cambiato l'orientamento del dispositivo.
iPhone chiederà all'applicazione se vuole ruotare inviando un `shouldAutorotateToInterfaceOrientation:` (in realtà questa descrizione non è del tutto corretta tecnicamente, ma per ora è utile per capire come vanno le cose, entreremo nei dettagli nel Capitolo 4). Rispondete `YES` o `NO` a questo metodo. Con `YES` iPhone ruoterà l'applicazione secondo l'orientamento del dispositivo, con `NO` l'applicazione non ruoterà. Ecco il codice per questo primo metodo:

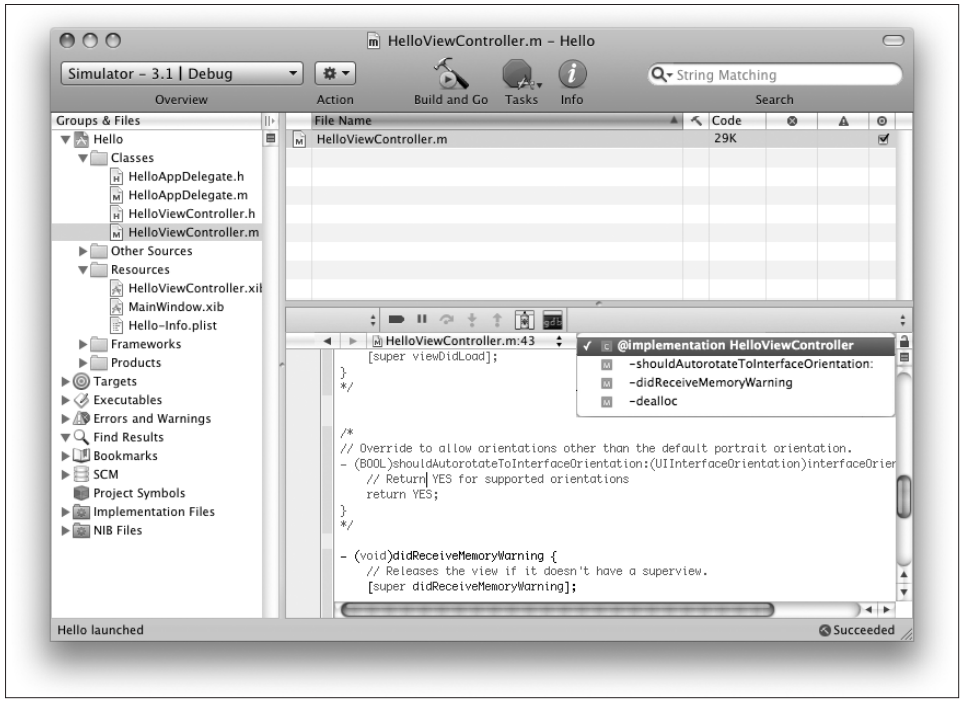
HelloiPhone/Hello/Classes/HelloViewController.m

```
- (BOOL)shouldAutorotateToInterfaceOrientation:
        (UIInterfaceOrientation)interfaceOrientation {
        return YES;
}
```

Poiché per questa applicazione volete supportare tutti gli orientamenti, dovete semplicemente restituire `YES`.
Abbiamo aggiornato l'interfaccia utente e il codice, perciò è giunto il momento di eseguire l'applicazione: fate clic sul pulsante *Build and Go* in Xcode per avviarla. Ad applicazione avviata, utilizzate le combinazioni di tasti Mela+Frecce per far ruotare il dispositivo nel simulatore. L'etichetta rimane centrata nella view e cambia il suo orientamento insieme al dispositivo.

2.6 Navigazione nel codice

In questo esempio non c'è molto codice, ma con progetti più complessi è utile sapere come navigare rapidamente all'interno dei codici sorgente con Xcode. Quando si seleziona un file nell'elenco, lo si apre nel pannello dell'editor; questo pannello presenta diversi pulsanti nella parte superiore che forniscono strumenti per navigare nel codice. Fate clic sul menu a discesa dei simboli per raggiungere immediatamente qualsiasi simbolo. Con un semplice clic sull'elenco dei simboli si visualizzano i simboli nell'ordine in cui sono riscontrati nel file. Se ritenete che sia più facile trovare ciò che state cercando in un elenco ordinato alfabeticamente, potete tenere premuto il tasto Opzione e fare clic sull'elenco dei simboli per attivare tale ordinamento.



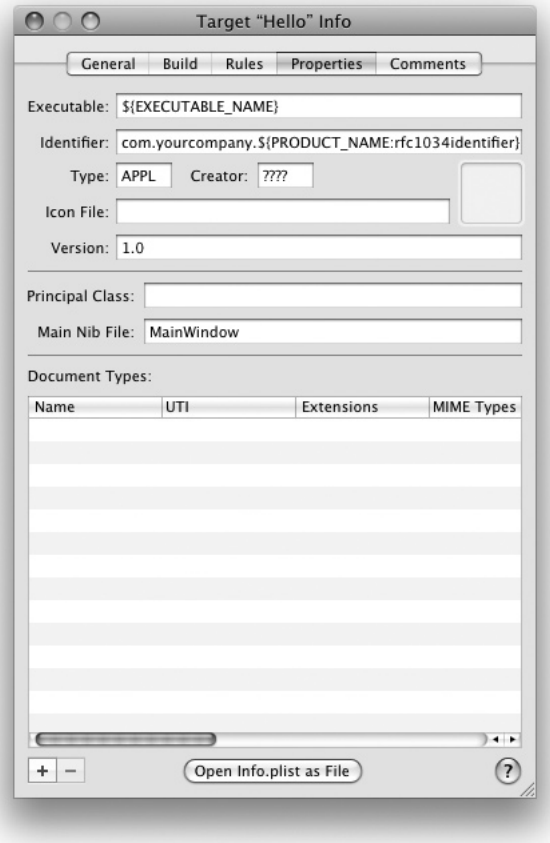
2.7 Esecuzione dell'applicazione su iPhone

Il simulatore è utile per fare delle prove, ma nulla può sostituire l'esecuzione diretta su un esemplare reale di iPhone, inviandovi il codice e osservando come reagisce nelle proprie mani. Sfortunatamente non potrete inviare il codice al vostro iPhone finché non avrete ricevuto i certificati di sviluppatore da Apple. Se avete già installato i vostro certificati, nel prosieguo di questo capitolo vedrete come installare la vostra applicazione nel dispositivo; se non li avete ancora installati dovete collegarvi al sito iPhone Developer Program, fare richiesta e passare alla sezione dove è spiegato come ottenere i certificati, installarli nei dispositivi e utilizzarli per firmare il codice da Xcode. Se non avete a disposizione i certificati, passate direttamente al prossimo capitolo.

La modifica da effettuare al progetto in Xcode per installare l'applicazione in iPhone è minima: per prima cosa dovete specificare l'identificatore di applicazione appropriato nelle proprietà del progetto. Selezionate il progetto e premete Mela+i (o fate clic con il pulsante destro del mouse e scegliete *Get Info*) e poi selezionate la scheda *Properties*.

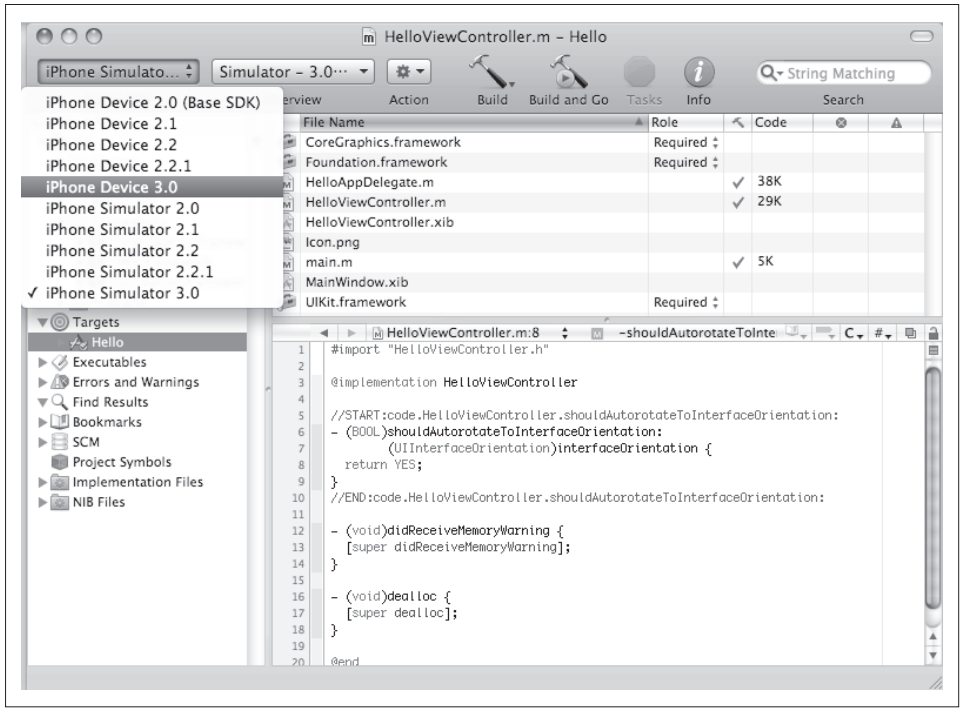
Nella figura a pagina seguente si vede la configurazione utilizzata per questo esempio:

L'identificatore utilizzato deve corrispondere a uno dei profili impostati con i vostri certificati di sviluppatore iPhone. Personalmente ho creato un profilo specifico da utilizzare con gli esempi di questo libro, e vi suggerisco di fare altrettanto. Se non volete creare un profilo specifico, potete utilizzare un identificatore corrispondente a un profilo esistente.



Ora dovete passare dall'esecuzione nel simulatore a quella su iPhone. Selezionate il menu a discesa *Overview* (come mostrato nella figura a pagina accanto). Nell'elenco *Active SDK* scegliete *Device -- iPhone OS X* (dove *X* è la versione effettivamente installata nel dispositivo). Se non vedete questa voce, probabilmente avete un problema con i vostri certificati di sviluppatore; collegatevi al sito di Apple dedicato agli sviluppatori per iPhone e consultate le indicazioni per verificare di aver seguito correttamente la procedura necessaria per installare i certificati e utilizzarli.

Una volta selezionato il dispositivo, fate clic nuovamente sul pulsante *Build and Go*. La procedura richiederà qualche secondo perché Xcode deve ricompilare tutto il codice per renderlo eseguibile sul processore di iPhone anziché su quello del Mac. Una volta completata la compilazione, la creazione del pacchetto e la firma dell'applicazione, Xcode la installa nel vostro dispositivo e la avvia. Una volta avviata, provate a ruotare il vostro iPhone e osservate come il testo 'Ciao iPhone' ruoti in modo corrispondente.



Congratulazioni, avete completato la procedura per compilare ed eseguire la vostra prima applicazione iPhone. Avete visto un po' di Xcode e IB e come utilizzare questi due strumenti per realizzare codice e interfaccia delle applicazioni.