前言

歡迎閱讀本書!這本書將藉由實作一個功能豐富的 iOS 應用程式,來引導讀者學習 Swift 程式語言,這個應用程式的功能包括存取相機、操作檔案系統、臉部辨識以及操作 iOS 的圖形系統。在過程中,我們也會向你介紹 Swift 程式語言最新最先進的各種主題。

Swift 是一個令人驚艷的現代程式語言,汲取其他新程式語言的優點,讓它成為一個易 寫、易讀且不易犯錯的程式語言。

我們覺得學習 Swift 最好的方法就是實作 app ! 實作 app 就需要 framework,而最好的 framework 之一,就是 Apple 提供的 Cocoa Touch,它能用來建立高品質、功能強大的 軟體。本書的書名也可以改為《用 Swift 學習 Cocoa Touch》,因為 framework 的使用和 Swift 語言具有同等的重要性。本書撰寫時 Swift 是第 4 版,而且它的前途持續看好。

程式碼範例

我們建議你逐章撰寫自己的程式碼,不過如果你碰到困難或是想要保存範例程式碼的話,你也可以在我們的網站上找到範例程式碼(*http://www.secretlab.com.au/books/learning-swift-3e*)。

適用讀者

本書重點放在 Swift 4,並不會討論到 Objective-C 的部分。雖然我們可能會在書上提到 它幾次,但我們並不預設你知道怎麼使用它。在書的開頭,我們會說明 Swift 4 的基本



概念,然後接下來藉著在 iOS 上建構一個具拍照功能的 Selfiegram app,盡可能地利用 它做語言以及 Cocoa Touch 的使用教學。

本書與你之前看過的書所追求的目標或有不同,如前面提到過的,我們相信最好的學習 方法就是透過實作 app 來學習 Swift。我們假設讀者是會寫程式,對 macOS 和 iOS 的 使用環境感到熟悉,但不假設你有使用 Swift 或 Objective-C 在 iOS 上做過程式開發的 經驗。

本書架構

我們將在本書中說明 iOS 的 framework: Cocoa Touch,也會教你如何使用 Swift 的語法 與功能。

第一部分 "歡迎使用 Swift",我們會看到用來開發的工具,還有 Apple 開發者計劃 (Apple Developer Program)。接著說明 Swift 程式語言的基本概念,並在 Apple 平台上 建立一個程式,以及一般程式的架構組成。

第二部分 "建立 Selfiegram",我們會開始建立 Selfiegram,它是一個 iOS 上的照相應用 程式,也是本書的重心所在。在這部分中,我們會建立重要的基礎功能,像是拍攝一張 照片並儲存在磁碟中。

第三部分 "為 Selfiegram 增加功能",我們會為它加入一些有趣的功能,像是自訂相機 視窗、臉部偵測、存取網路以及為該 app 加入佈景主題。

第四部分"**開發** Selfiegram 之外",我們將會討論一些功能和工具,它們可以幫助你未來的開發工作,包括如何使用 Xcode 除錯以及側寫工具,以及一些可讓你人生更輕鬆地 第三方工具。

本書編排慣例

本書使用以下的編排規則:

斜體字 (Italic)

代表新的術語、URL、電子郵件地址、檔案名稱及副檔名。中文以標楷體表示。



第一章

開始

歡迎閱讀本書!在這一本書中,我們要帶領完全不懂 Swift 怎麼寫的你,去寫出一個 iOS 11 適用的多功能 app。在過程中,我們會探索函式庫、framework 以及身為 iPhone、iPad 軟體開發者的你所有可用的功能。我們會為常見的困難提出實際的解決方法,也會詳細的講述 Swift 語言的使用。

在你開始使用開發工具之前,我們要先做一些重要的工作,特別是要先瞭解 Apple 的開發者計劃,並讓你取得你的帳戶,如此一來你才能在裝置上建立你的 app。



本書假設你已有兩樣東西:一台 Mac 以及一個 iOS 裝置(例如 iPhone 或 iPad)。

這兩樣東西之中, Mac 是必要的;如果沒有 Mac 的話,你無法執行 Xcode, Xcode 是用來寫程式碼、設計介面還有建置與執行你程式的工 具。你的 Mac 需要能執行 Xcode 9.2 或更新版本,所以它必須要使用 macOS 版本 10.12 以上。

而 iOS 裝置不是必要,但如果沒有它的話,由於本書所要做的 app 使用 了硬體功能,而這個功能不存在 iOS 模擬器中,所以會導致你無法跟著做 完本書所有的內容。你的 iOS 裝置必須搭配 iOS 11 或更新的版本。



Xcode

Xcode 是 Apple 平台上所有開發工作的開發環境。當你閱讀這本書時,你大部分時間都 會使用 Xcode,所以本章後面所有的內容都在說明如何取得 Xcode、安裝以及熟悉它的 介面。

取得 Xcode

Xcode 是從 App Store 上取得,可以在 App Store 中搜尋 Xcode 或到 Apple 的開發者 下載網頁(*https://developer.apple.com/download/*)中找到 Xcode,並點擊 Download 按鈕。



如果不喜歡使用 App Store,你可以直接從 Apple 的 Apple Developers 中 Downloads 頁面(*https://developer.apple.com/download/more/*)下載, 這個網頁上列有所有的 Xcode 版本,確認你下載的是最新版即可。

裝好了以後,就直接執行它吧。你會看到如圖 1-1 的歡迎畫面。



圖 1-1 Xcode 的歡迎畫面



建立你的第一個專案

由於我們要介紹 Xcode,所以會從建立一個完全空白的專案開始(我們在第4章建立本書 app 時又會再做一次):

1. 點擊 "Create a new Xcode project", 然後會跳出樣板選擇(見圖 1-2):

Application	a macoa cioas-p	anom	10	AI HE
1	*	AR		•
Single View App	Game	Augmented Reality App	Document Based App	Master-Detail App
			\bigcirc	
Page-Based App	Tabbed App	Sticker Pack App	iMessage App	
Framework & Libra	ry			
		N		
Cocoa Touch Framework	Cocoa Touch Static Library	Metal Library		

圖 1-2 Xcode 樣板選擇

2. 選擇 Single View App,並點擊 Next。

Xcode 會要求你為新的 app 提供一些說明描述,由於你現在只是藉由建立這個 app 來熟悉 Xcode 的使用,並不是真的要用它來寫程式,所以隨意在欄位中填寫什麼都 無所謂,不過在第4章時就不會隨意填了。

完成了以後,點擊 Next,並選擇要存放專案的地點。
現在你看到的是 Xcode 的主要介面,如圖 1-3。



	D 191 In HelloSwift		
Y O HelbSelft	Director Capab	lities Resource Tags Info Build Settings Build Phases	Build Rules Identity and Type
Headsburt Acadeulegala.switt VeedContrains.eet WeedContrains.eet WeedContrains.eet Mean avgorder Maassa sosaats Acadeulegala.switt Meansatures Meansatures Meansatures Meansatures Meansatures	PROJECT TANGOT TANGOT A Jennicowiti		News Heinzberfr Loottor Heinzberfr Verlagterfr Heinzberfr Verlagterfr Heinzberfr Project Desverst Heinzberfr Displaceation Othery Media Displaceation Othery Media Displaceation Othery Media Displaceation State Previous Wess A I I I I I I I I I I I I I I I I I I I
		* Deployment Info	DUGD
		Deployment Target	No Matches
+ 100	1+-3-	Upside Down	38 (Q)

圖 1-3 Xcode 主要介面

Xcode 介面

Xcode 用一個視窗,將整個專案資訊顯示給你看,這個視窗被切為數個區域,你可以依 想看的內容任意打開或關閉這些區域。

讓我們來看一下這些區域,並瞭解它們是用來幹嘛的。

編輯器

Xcode 的編輯器會是你最常用到的區域,所有的程式碼編輯、介面設計以及專案設定等都在這個區域,它會依你打開的功能顯示不同的內容。

如果你編輯的是程式碼,編輯器就變成文書編輯器,這個編輯器帶有程式碼自動補完、 語法提示,以及所有一般開發該有的功能。如果你想要修改使用者介面檔案,此時編 輯器就會變成視覺編輯器,讓你可以拖曳並放置想用的介面元件。編輯其他種類的檔案 時,編輯器也會變成各種對應的樣子。

在新建專案時,編輯器會顯示的是專案設定,如圖1-4。



General Capa	bilities Resource Tags	Info	Build Settings	Build Phases	Build Rules
PROJECT	▼ Identity				
TARGETS	Display Name	HEIDSAM			
A HelloSwift	Bundle Identifier	com oreilly le	arainacwift HelloSu	ife.	
	Buildie identitier	controlenty.ie	anningswirtenoow		
	Version	1.0			
	Build	1			_
	Team Provisioning Profile Signing Certificate	Automatica Acode will a certificates Jonathon Ma Xcode Manage iPhone Develo	Illy manage signing reate and update pro unning (Personal Te ad Profile () per	files, app IDs, and am)	
	* Deployment Info				
	Deployment Target	052			
	Devices	Universal		1	
	Main Interface	Main			

圖 1-4 顯示專案設定的 Xcode 編輯器

編輯器也可以將畫面拆成 *main editor*(主要編輯器)和 *assistant editor*(輔助編輯器)。當 main editor 開啟檔案時, assistant editor 會顯示相關檔案。它會一直隨著 main editor 當時所開啟的檔案,去切換顯示的相關檔案。

舉例來說,如果你開啟了一個介面檔,此時 assistant editor 就會顯示所有這介面相關的 程式碼。如果你開啟另外一個介面檔,那麼 assistant editor 也會隨之更新它的內容。

在編輯器的上面,你會看到一個 jump bar, jump bar 的功能是讓你從正在編輯的內容, 快速跳到另外某個相關內容,如:跳到同一個目錄中的其他檔案。使用 jump bar 是瀏覽 專案的一個快速方法。



工具列

Xcode 的工具列(如圖 1-5)的角色像是整個介面的任務控制者,在你開發應用程式的 過程中,它是 Xcode 中最不會變來變去的了,它提供了一個讓你控制你的程式動作的 地方。

🔹 🔹 🕨 🔹 🗛 Helostimit) ஞ effort K. Helostimit) 📾 effort K. Helostimit) 📾 effort K. Helostimit 2333 pm

圖 1-5 Xcode 的工具列

從 macOS 視窗控制的右邊開始,工具列從左到右的功能是:

執行鍵(圖1-6)

按這個鍵叫 Xcode 編譯並執行應用程式。

依所選的應用程式類型,以及你目前選擇的設定不同,該鍵會有不同的功能:

- 如果你建立的是 Mac 應用程式,新的 app 會出現在 Dock 裡,並且在你的機器上 執行。
- 如果你建立的是 iOS 應用程式,新的 app 會在 iOS 模擬器或是已連結的 iOS 裝置(如 iPhone 或 iPad)上執行

此外,如果你按住這個鍵不放,可以把它從執行改為其他功能,例如:測試、側寫 或分析。測試功能可執行任何你預先準備好的單元測試;側寫則會執行 Instruments (我們在第 17 章會提到)應用程式;而分析功能則會執行程式碼檢查並指出有潛在 問題的地方。





停止鍵(圖1-7)

按下這個鍵會讓 Xcode 停止任何正在進行的動作 —— 比方正在建置應用程式的話, 就會停下來,如果正在執行除錯器的話,也會離開除錯。





圖 1-7 停止鍵

Scheme 選擇(圖 1-8)

Xcode 中的建置設定稱為 *scheme*,也就是要建置什麼、如何生成以及在哪裡執行程式(例如:在你的電腦上或是連結的裝置上執行)。

一個專案可以包含多個 app,可以從 scheme 選擇去切換想建置的目標是哪一個 app。

如果想要切換建置目標,從 scheme 的左側按下去即可。

你也可以設定應用程式執行的位置,如果你建置的是一個 Mac 應用程式,你只會在 Mac 上面執行。如果你建置的是 iOS 應用程式,可以選擇在 iPhone 或是 iPad 模擬器上執行(基本上是同一個應用程式,它只是會照你所選的 scheme 改變外形而已)。你也可以選擇在已連結設定好的 iOS 裝置上執行,這部分在第 14 頁"執行你的程式碼"中會有更多討論。



圖 1-8 scheme 選擇器

狀態顯示(圖1-9)

圖 1-9 狀態顯示

顯示 Xcode 正在進行著什麼工作——例如正在建置你的應用程式、下載文件或是安裝應用程式到連結的裝備等等。

如果有多個工作同時進行,左側會出現一個小按鍵,可以輪流切換不同的工作狀態。



· 名峯 www.gotop.com.tw

編輯器選擇器(圖1-10)

可以用來決定編輯器顯示的樣子,你可以選擇單一編輯視窗、加上輔助視窗或變成版本編輯器。在你有使用版本控制系統(像 Git 或 Subversion)時,版本編輯器可以讓你比對檔案兩個不同版本間的差異。



圖 1-10 編輯器選擇器



在本書中並不會談到版本控制系統,但它是一個很重要的課題。我們推薦 讀者可以參考 Jon Loeliger 與 Matthew McCullough 的《*Version Control with Git*, 2nd Edition》(O'Reilly 出版)。

View 選擇器(圖 1-11)

透過 view 選擇器控制瀏覽、除錯與 utilities 畫面要不要顯示。如果你覺得畫面太過 擁擠想要畫面乾淨些,可以快速的從這裡開關顯示。



圖 1-11 view 選擇器

Navigator

Xcode 視窗的左側是 navigator,它會顯示你目前專案的資訊(圖 1-12)。



	묘	Q	\triangle	Θ		Ę
🔻 📓 He	lloSwift					
•	HelloSv	vift				
	App	Deleg	ate.sw	ift		
	View	Conti	oller.s	wift		
	Main	.story	board	E.		
	Asse	ts.xca	assets			
[Laun	chSc	reen.s	torybo	oard	
1	Info.	olist				
1 R	Product	s				

圖 1-12 Xcode 的 navigator 窗格

navigator 裡有 8 個分頁, 自左而右分別是:

```
Project navigator(專案瀏覽)
```

是最常使用的 navigator, 它可列出專案裡所有檔案, 點選即可顯示在編輯器中。

Symbol navigator (符號瀏覽)

列出專案中所有類別與函式,如果你需要顯示所有類別,並快速進到某個類別的方 法中時特別好用。

Search navigator (搜尋瀏覽)

讓你在專案裡找特定的文字。(快速鍵是 第-Shift-F,若按 第-F 則是搜尋目前開啟的 文件。)

Issue navigator(問題瀏覽)

列出 Xcode 認為你的程式碼有問題的地方,包括警告、編譯錯誤與內建程式碼分析 器認為有問題的地方。



Test navigator (測試瀏覽)

顯示所有專案中的單元測試,單元測試以前只是 Xcode 的選配元件,現在已經是預設內建的元件了,在第130頁的"測試 SelfieStore"小節中會介紹更多相關內容。

Debug navigator (除錯瀏覽)

在你執行除錯時才能用,它可讓你檢視專案中不同執行緒的狀態。

Breakpoint navigator (中斷點瀏覽)

列出所有你在除錯時所設定的中斷點。

Report navigator (報告瀏覽)

列出 Xcode 對專案做的動作結果報告(例如建置、除錯及分析)。你也可以查看之前 Xcode 進行建置的報告。

Utilities

對於正在編輯器中進行的工作,utilities 窗格(圖1-13)可顯示更多的額外資訊。舉例 來說,如果你正在編輯一個 Swift 程式碼檔案,utilities 窗格可以讓你檢視與修改該檔案 的設定。

utilities 窗格又分為兩個區域:*inspector*(檢視器)與*library*(元件庫)。inspector可以顯示更多的所選項目資訊;而*library*是你可以加入專案的元件集合。這兩個工具在你設計使用者介面時一直會用到;而且,*library*中還有其他一些有用的東西,例如檔案樣板或程式碼片段,可以直接拖曳使用。



Name AppDelegate.swift Type Default - Swift Sour Location Relative to Group	
Type Default - Swift Sour	-
Location Relative to Group	e 🗘
Ann Dologoto quift	0
AppDelegate.switt	in the
Full Path /Users/desplesda/ Downloads/HelloSwif HelloSwift/ AppDelegate.swift	0
On Demand Resource Tags	
Only resources are taggable	
Target Membership	
HelloSwift	
Text Settings	
Text Encoding No Explicit Encoding	0
Line Endings No Explicit Line End	ngs 🗘
Indent Using Spaces	0
Widths 4 C Tab Ir Wrap lines	4 Ç dent
Cocoa Touch Class - A Co Touch class	coa
UI Test Case Class - A cla implementing a unit test	S-
Unit Test Case Class - A implementing a unit test	lass
	-

圖 1-13 utilities 窗格正顯示一個程式碼檔案的資訊



除錯區

在程式執行時,除錯區(圖 1-14)會顯示除錯器回報的資訊。你不論何時想看應用程式 回報了什麼資訊,都可以在這個區域看到,預設在程式執行時才會出現除錯區。

圖 1-14 除錯區

這個區域又分為兩個部分,應用程式被暫停時,左邊的區域顯示本地變數及其值,右邊 則顯示除錯器即時生成的訊息,其中也包括了應用程式回報的訊息。

你可以藉由點擊視窗右上方的 view 選擇器,來開啟或關閉除錯區(圖 1-15)。

|--|

圖 1-15 view 選擇器中央的按鈕,可以隱藏或顯示除錯區

執行你的程式碼

當你建置一個 iOS app 時,你寫的程式碼並不會在你正用來開發的電腦上執行,你的程 式碼是被設計來跑在 iOS 上的,這表示若你想測試你的程式碼的話,有兩種選擇:在 iOS 模擬器上執行它,或是在一個實際的裝置上執行它。

iOS 模擬器

由於 iOS 模擬器(simulator)不需要任何設定就可以使用,所以我們先從 iOS 模擬器開始說明。



Xcode 附帶許多的模擬器,每一個都是為了模擬支援中或最新的 iPhone 和 iPad 的特性 而設計的。

Xcode 內建數種模擬器,打開 scheme 選擇器並從列表中選擇一種模擬器(見圖 1-16)。 點擊 Build 按鈕,或按下 Command-R 進行建置;待建置完以後, app 就會在模擬器中被執行了。

Device Kodiak	
- Roulak	
Build Only De	vice
/ Generic	iOS Device
IOS Simulator	s
📑 iPad (5t	h generation)
📑 iPad Air	
📑 iPad Air	2
iPad Pro	9.7-inch)
iPad Pro	o (10.5-inch)
iPad Pro	o (12.9-inch)
iPad Pro	(12.9-inch) (2nd generation)
iPhone	5s
📑 iPhone	6
📑 iPhone	6 Plus
📑 iPhone	6s
📑 iPhone	6s Plus
iPhone	7
iPhone	7 Plus
📑 iPhone	8
iPhone	8 Plus
iPhone	SE
🗸 📑 iPhone	x
Add Additiona	I Simulators
Download Sim	nulators

圖 1-16 Xcode 9.2 中內建的模擬器列表



模擬器和真實的裝置有幾個很大的差別,例如,模擬器(simulator)和硬體模擬器(emulator)是不一樣的東西;使用模擬器時,你的程式碼是由你 Mac 中的 Intel 晶片所執行,而在 iOS 裝置上則是由 ARM 晶片執行。

這個差異點代表,在模擬器上執行 app 的效能,和在實際裝置上執行的效能有落差。而 且,通常模擬器會有更多的記憶體可用,如果你的 app 在模擬器上能夠正常執行,並不 代表在真實的裝置上也會有一樣好的效能表現。

最後,模擬器缺少數種真實裝置的硬體功能,舉例來說,模擬器沒有相機,相機對我們 在書中將要製作的 app 來說是非常重要的功能。

在實際裝置上執行

scheme 選擇器可以讓你選擇一個已連結的 iOS 裝置,以它為目標進行建置並在它上面執行你的 app。在一個真實的裝置上執行程式幾乎和在模擬器一樣地簡單,只差在一點,就是要在實際裝置上跑的程式碼,必須要經過開發者認證簽章。

開發者認證是用來識別寫程式碼的人或公司的東西,iOS 裝置只允許執行簽章過的程式碼,而用來簽章的證書必須經過 Apple 公司簽屬。

當你登入用來開發的 Apple ID 時,Xcode 就會自動地為你產生認證,你只要做完這幾步,就可以登入 Apple ID:

- 1. 打開 Xcode 選單,選擇 Preferences
- 2. 到 Accounts 分頁,點擊+按鈕
- 3. 登入你的帳戶

一旦登入帳戶後,你要將你的 app 和一個團隊(*team*)做關聯。團隊是 Xcode 用來組織 開發者和其認證的方法;如果你是獨立一人開發的話,你的團隊就只會有你自己一人。

要設定關聯,請做以下步驟:

- 1. 在專案 navigator 中選擇最上層的專案(在最上面的項目,用藍色文件圖示標示的那個)。
- 2. 在 main editor 中的 Signing 節區,自 Team 下拉選單中選擇一個團隊。

接下來 Xcode 會為程式碼簽章做一些動作,結束後就可以開始在裝置上執行你的 app 了。



當你把一個專案和一個團隊做關聯時,如果你尚未有任何認證,也沒有任何設定文件(provisioning profile)的話(provisioning profile 是用來將 app 連結到你的認證用的一種檔案),此時 Xcode 會自動地為你產生一個 認證。若不想讓 Xcode 幫你做這件事的話,你也可以手動透過 Apple 開發 者網站(https://developer.apple.com/account/ios/certificate/)處理。我們 會在第 333 頁的"用 match 做程式碼簽章"小節中討論更多細節內容。

要建置並執行你的 app,請做以下動作:

1. 確認你的裝置已連接到你的 Mac。

2. 自 scheme 選擇器中選取你的裝置,並點擊 Run 按鈕(或按下 Command-R)。

發布你的 App

在你自己的裝置上執行程式碼是免費的,除了 iPhone 或 Mac 安裝時就設定好的 Apple ID 之外,也不需要再登入任何東西。

不過,如果你想要發布你的 app 給別人,例如透過 Apple 的 TestFlight 或 App Store 給別人的話,你就必須加入 Apple 開發者計劃(Apple Developer Program)(*https://devel oper.apple.com/programs/*)。在寫書之時,加入開發者計劃的費用是每年 \$99 美元,同時可存取像 TestFlight 以及 iTunes Connect 這樣的工具, TestFlight 讓你將 beta 版的 app 發布給測試者, iTunes Connect 則是一個讓你可以傳送 app 到 App Store 的控制台。



若只是想跟著本書的進度建置範例 app 的話,是不需要參加 Apple 開發者 計劃的,但對於第 19 章和第 20 章的內容,就必須要參加了。

介面建立器

除了寫程式之外,Xcode 也是用來設計你 app 使用者介面的應用程式,對於 iOS app 來 說,介面設計和程式碼都很重要,所以我們會花一些時間來練習使用介面建立器。

當你從專案 navigator 選取一個 storyboard 檔案後,就可以叫出介面建立器了。現在先選擇 Main.storyboard 檔, Xcode 會如圖 1-17 般叫出介面建立器。





圖 1-17 介面建立器

關於介面建立器,還必須要知道幾個重要的功能:

- 在編輯器的左側,你可以找到 outline,它含有該文件中的所有物件,同時也會顯示這些物件的結構。
- 在 Xcode 視窗的右側,會被調整成可以顯示該介面中物件的屬性。此外,在 *library* (在 utilities 窗格中的底部),有可以用來建置介面的 UI 元素列表:你可以藉由點 擊物件庫(Object Library) 按鈕,來取得這個列表。物件庫按鈕是 library 窗格上 方,從右方開始數第二個,它的圖示上有圓圈標示(見圖 1-18)。





圖 1-18 物件庫 (object library)

iOS 中的介面

在 iOS 中只要是看得見的東西都稱為 view,也就是說,只要在畫面上看得見或可以和使用者互動的任何東西包括按鈕、標籤和文字欄位等,都是 view。

view 也可以內嵌在其他 view 裡面,你在裝置上看到的每個"畫面"內容,都是一個 view,這個概念是 iOS 應用程式基本的設計概念。

這個最上層的 view 是由一個 view controller 物件所管理的;若要弄出一個新的畫面,你 需要建立 UIViewController 的子類別物件,這個子物件會覆寫畫面生命週期中會呼叫的 函式,而且也包含你建立畫面的功能所需要使用的方法。另外,你需要在介面建立器中 準備 view controller 需要使用的介面,並且將該介面連結到你已加好的類別。

在本書的教學之中,我們會很常切換到介面建立器中,所以對 iOS app 介面有這個的基本瞭解是很重要的。



本章總結

在這一章中,我們安裝好並看過一輪用來建立 app 的工具——Xcode。在下一章中,我 們要來討論 Swift 程式語言。

