

# 致理技術學院

## 資訊網路技術系

### 實務專題報告

#### iPhone 兒童字母學習 APP

指導教師： 蕭勝華

學生：王碩志(69634120)

張志豪 (69634105)

樂愷 (69634143)

江俊杰(69634102)

劉俊麟(69634135)

中華民國 101 年 05 月

# 致理技術學院

## 資訊網路技術系

### 實務專題報告

#### iPhone 兒童字母學習 A P P

學生：王碩志(69634120)

張志豪 (69634105)

樂愷 (69634143)

江俊杰(69634102)

劉俊麟(69634135)

本成果報告書經審查及口試合格特此證明。

指導教師：\_\_\_\_\_

中華民國 101 年 05 月

## 專題研究授權書

本授權書所授權之專題研究為\_\_\_\_\_

共\_\_\_\_\_人，在致理技術學院資訊網路技術系 \_\_\_\_\_學年度第\_\_\_\_\_學期  
完成資網實務專題。

專題名稱：\_\_\_\_\_

\* 同意  不同意

本組同學共\_\_\_\_\_人，皆同意著作財產權之論文全文資料，授予教育部指定送繳之圖書館及本人畢業學校圖書館，為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種方法重製，

不限地域與時間，惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未勾選，該組同學皆同意視同授權。

指導教師姓名：

專題學生簽名：

學號：

(親筆正楷)

(務必填寫)

中華民國 \_\_\_\_\_年 \_\_\_\_\_月 \_\_\_\_\_日

# 誌 謝

在致理這幾年的時間裡，我們必須去面對跟以往高職或者五專沒有學習過的技能，在有限的時間內去完成一個專題，再加上我們是夜校大家上下班的時間都不一致，無疑是一項不可能的任務。在開始籌畫時我們遲遲無法下定題目，經過不斷的討論之後總算找到了一個方向，但是找到一個方向並不困難，困難的是在於不知從哪開始下手，一個手機APP看似簡單，但裡面暗藏著許多技術以及架設概念，我們就像剛出社會的少年要去面對這廣闊無邊的世界，這也使我們的專題進度十分緩慢。

還好在漫無頭緒的摸索中尚有指導老師不辭辛苦的指導、提點。我們也漸漸的找到遵循的方向，做專題也讓我們理解網路世界有多麼便利，想要什麼資訊都可以去網路搜尋的到，在這期間從書上、網路所獲取的資訊以及技巧、經驗絕對是讓我們彌足珍惜的，想起一開始交給指導老師那漏洞百出的程式以及在我們眼內所謂的正式書面報告都無法讓指導老師的眉頭如鮮花盛開般舒放，還好在指導老師不斷的督促以及召集我們討論講解還有專業書本的支援中，總算有了現在的成績。在此，非常感謝我們的指導老師蕭勝華，在公務繁忙時仍抽身為我們指導專題，並且給我們充分的時間去討論和理解，讓我們順利完成這份畢業專題。

# 摘 要

現今的社會，手機發展極為迅速，手機的功能也越來越方便，以至於手機 A P P 下載軟體成為越來越受歡迎的消費模式，加上手機 A P P 不限時間和空間限制，不用在家裡就能購買自己想要的軟件，並且不用電腦下載或是去實體店面去購買非常的方便，和傳統實體方式不同的是，只要點一下，輸入帳號，讓手機自動下載安裝，所以手機 A P P 購買軟體已成為時下不可或缺的購物方式。

技術方面使用到了 Xcode 來設計軟件商品，MAC os X 作業系統的應用。

**關鍵詞：** MAC OS X，Xcode

# 目 錄

第一章 緒論 .....	1
第一節 前言 .....	1
第二節 研究動機與目的 .....	2
第二章 相關研究應用探討.....	7
第三章 系統原理與分析.....	17
第一節 系統架構 .....	17
第二節 系統環境 .....	18
第三節 使用軟體： .....	23
第四節 系統特色 .....	29
第五節 進度規劃 .....	30
第六節 人力分配 .....	31
第四章 專題實作.....	32
第五章 系統操作.....	34
第一節 系統環境安裝 .....	34
第二節 系統操作說明 .....	56
第六章 結論 .....	60
參考文獻 .....	61

# 圖目錄

圖 1 基本元素 .....	6
圖 2 模式一 .....	7
圖 3 模式二 .....	8
圖 4 模式三 .....	9
圖 5 模式四 .....	10
圖 6 模式五 .....	11
圖 7 模式六 .....	12
圖 8 模式七 .....	13
圖 9 模式八 .....	14
圖 10 模式九 .....	15
圖 11 A P P 與開發者關聯.....	16
圖 12 iphone 產品圖 .....	17
圖 13 iPhone 作業系統應用技術層 .....	18
圖 14 MAC OS X .....	22
圖 15 OS X 架構圖 .....	23
圖 16 Xcode .....	25
圖 17 Photoshop .....	26
圖 18 安裝畫面--1 .....	33
圖 19 安裝畫面--2 .....	34
圖 20 安裝畫面--3 .....	34
圖 21 安裝畫面--4 .....	35

圖 22 安裝畫面--5.....	36
圖 23 安裝畫面--6.....	37
圖 24 安裝畫面--7.....	38
圖 25 安裝畫面--8.....	39
圖 26 安裝畫面--9.....	39
圖 27 安裝畫面--10.....	40
圖 28 安裝畫面--11.....	41
圖 29 安裝畫面--12.....	42
圖 30 安裝畫面--13.....	43
圖 31 Xcode 註冊畫面--1.....	44
圖 32 Xcode 註冊畫面--2.....	45
圖 33 Xcode 註冊畫面--3.....	45
圖 34 Xcode 註冊畫面--4.....	46
圖 35 Xcode 註冊畫面--5.....	47
圖 36 Xcode 註冊畫面--6.....	48
圖 37 Xcode 註冊畫面--7.....	52
圖 38 Xcode 註冊畫面--8.....	50
圖 39 Xcode 註冊畫面--9.....	51
圖 40 Xcode 安裝畫面--1.....	52
圖 41 Xcode 安裝畫面--2.....	52
圖 42 Xcode 安裝畫面--3.....	53
圖 43 Xcode 安裝畫面--4.....	61
圖 44 Xcode 安裝畫面--5.....	61

圖 45 Xcode 安裝畫面--6 .....	55
圖 46 Xcode 安裝畫面--7 .....	55
圖 47 Xcode 安裝畫面--8 .....	56
圖 48 Xcode 安裝畫面--9 .....	57
圖 49 系統操作說明--1 .....	58
圖 50 系統操作說明--2 .....	58
圖 51 系統操作說明--3 .....	59
圖 52 系統操作說明--4 .....	60
圖 52-1 系統操作說明 首頁 .....	60
圖 53 系統操作說明--5 .....	61
圖 54 系統操作說明--6 .....	61
圖 55 系統操作說明--7 .....	62
圖 56 系統操作說明--8 .....	62
圖 57 系統操作說明--9 .....	63

# 表 目 錄

表 1 甘特圖 (Gantt Chart) .....	32
表 2 工作分配表 .....	33

# 第一章 緒論

## 第一節 前言

iPhone是搭載iOS行動作業系統，結合相機手機、個人數位助理、媒體播放器及無線通訊裝置功能為一體的手持設備，由蘋果公司執行長史提夫·賈伯斯於2007年1月9日舉行的Macworld發佈，2007年6月29日在美國上市。發表於2011年10月4日的最新一代iPhone「iPhone 4S」具有攝錄、攝影、發送簡訊、接收語音留言以及個人語音助理等功能，可以作為移動媒體播放器或者瀏覽網際網路和收發電子郵件使用，同時可使用高達數十萬的擴展應用程式增加功能，支持使用Wi-Fi和3G網路連接，預計未來將會支持4G LTE。iPhone的用戶界面是裝載在iPhone的Retina多點觸控螢幕上，於軟體內也內建虛擬鍵盤。第三方研發以及蘋果公司自行研發的眾多擴展應用程式集中在系統內建程式App Store中，從免費到付費皆有。

APP時代來了！據估計在今年市場規模可以到達150億美元，所有人都在看這麼龐大的新興商機如何分一杯羹。不過，如果你以為APP經濟的重點只在技術能力，那你就錯了。和其他的事業一樣，還是要先思考什麼是正確的商業模式，才會讓創意與技術發揮最大的商業價值。也許有人會問「App的商業模式不就是“收費”或“免費但附廣告”這兩種？」，當然不只這麼簡單。也許借用中國人老祖宗的說法「太極生兩儀，兩儀生四象，四象生八卦，八卦又可以化為八八六十四卦」。所以光是收費或免費，其實都還有種種衍生的組合。而這些不同的組合與變化，正是創業家與開發者得以由其中獲利的方式。

## 第二節 研究動機與趨勢

近來手機 A P P 大量崛起，各家廠商各有各的好，但是唯獨蘋果的 I P H O N E 最為熱賣，對於全球市場佔有率很大，現在在路上的大學生不誇張，幾乎人手一只，連我自己也不例外。

這次的專題作業研究是以開發手機軟體為主軸。早期手機大都是只有接電話或者是傳簡訊，再到後來又多了動態訊息或者一些加值服務，如聽股票或者是一些代訂票，現在的手機卻跟一臺電腦差不多了，現今的手機多了網路功能，手機開發的也越來越強大了，雙核三核四核，比比皆是，但是唯獨蘋果電腦所開發的這款手機乍看之下手機配備沒有比一些其他的廠牌好，但是他卻可以風靡全球，我想原因就在于他的方便，以及很多的應用軟體可以下載購買，而我們所要做的就是在他的 A P P 商店裡販賣我們所製作的軟體。

要如何才能讓自己在這個高倍數成長的 A P P 虛擬賣場中佔有一席之地？要如何把程式當自己的事業般經營？要如何讓人性化且便利的應用軟體帶來成就時掌握 A P P 的趨勢和變化，並擁有絕佳網路行銷優勢，與財富？在 A P P 賣場中，機會絕對是人人都有，不過要懂得把握絕竅才可能出奇制勝。

## 趨勢

### 行動裝置不是電腦

蘋果公司的平板電腦iPad一出，讚美與批評聲四起。讚美之詞在此就不提了，許多科技玩家的批評聲音集中在認為這是一款沒有達到期待的電腦，包含他的大小，價格，規格等等。這些聲浪充斥在Facebook，Twitter 以及眾多個人部落格上。

這似乎是一台很多科技玩家都不怎麼滿意的電腦，但這好像正是關鍵所在：以衡量電腦的標準，來衡量一台不打算被當成電腦的裝置。這個死結正是蘋果公司10年以來一直試圖告訴我們的。

是的，蘋果公司不是電腦公司，而行動裝置不是電腦，蘋果公司認為自己做出來的不是電腦。從2001年第一台iPod誕生開始，到2007年第一台iPhone，一直到2010年的iPad，這似乎是10年來Steve Jobs一直在告訴世人的事，而這件事是整個電腦產業一直沒有想通的。

近年來由於價格不斷大眾化，科技產品的使用族群發生了很大改變。對於CPU速度，記憶體容量等技術規格不敏感的消费群體，成了市場上新興的主流。他們對於行動裝置所展現出來的需求是多樣性的，個性化的，每個人要的都不一樣的。

此外，還產生了以應用為導向的認知與購買行為例如，能拿來聽音樂嗎？能拿來看電視嗎？能拿來看書嗎？這種以應用為導向的需求，導致了裝置更加的被細分到更分散的某一種小眾市場去。

## 過往的手段失靈

這些改變，是以英特爾和微軟為首的個人電腦產業不熟悉的。首先是單一商品賣出大量的時代已經過去。以前的電腦製造商頂多只要區分「商用機」和「家用機」，現在忙著應付「筆記型電腦」，「CULV輕薄筆記電腦」，「Netbook小筆電」，「平板電腦」等無數市場區隔。

更別說接下去可能還有MID（Mobile Internet Device）以及開始有電腦製造商跨入電子書閱讀器，智慧型手機的製造與銷售中。要管理這麼多的產品線，進而在銷售上迎合到各種消費者的需求，電腦製造商迎來的是一個「小量多樣」的時代。「大量少樣」的時代過去了。

其次，新興起的主流消費群體讓電腦製造商非常錯愕。之前定位在「個人的第二台筆記型電腦」的小筆電，顯然是打算賣給重度科技玩家，沒想到居然成為了許多人「人生中的第一台電腦」。這些消費者是學生，長輩，時尚人群。電腦產業用來做市場定位的那一套失靈了。

然而，製造商們似乎還沒有辦法繞出這個死胡同。回過頭來看看當蘋果推出iPad的時候，大家在想什麼？一些人認為，像iPad這樣的平板電腦，功能與規格以及能做的事情與智慧型手機相比差不多，如果這樣那還不如買小筆電。因此認為iPad沒有生存空間，不知道賣給誰。

這種比較法問題在哪？問題在於把iPad當成電腦。我們拿了一把量電腦的尺去量iPad，怎麼量都覺不對。然而，這種量法在過去兩年裡已被證明是死路一條：當小筆電問世的時候我們量了一遍，MID問世的時候又量了一遍，CULV問世的時候再量第三遍。我們找到答案了嗎？

## 以應用為導向，以消費者為核心

電腦製造商始終在這把電腦量尺上找不出答案。一開始小筆電被定位在筆記型電腦與智慧型手機的中間產品，後來筆記型電腦越做越小越便宜，侵入小筆電產品線。而小筆電的功能越做越強大，侵入到筆記型電腦市場。之後的CULV更是說不清楚市場定位，在三者間掙扎。

而還在概念階段只有小量產品的MID，號稱介於小筆電以及智慧型手機之間的市場區隔更是怎麼都說不清，電腦製造商感覺到自己怎麼做都不對。這些林林總總的問題，似乎都指向一個方向：這把量尺還合用嗎？我們拿這把尺繼續量下去是對的嗎？

把眼光從產品線定位拿開，移回消費者身上吧。這群對規格與技術不敏感的主流人群，他們到底要甚麼「應用」？讓iPod活起來的是音樂，讓iPhone活起來的是上網與上萬種Application 應用，現在蘋果打算拿電子書的應用讓iPad活起來。iPad不是電腦，蘋果認為不是。

蘋果這一次嘗試是否會成功還不知道。但是當Steve Jobs在iPad發布會現場提到蘋果公司時，說他是一家「行動裝置製造商」，我們知道他深深抓住了重點，那就是「行動裝置不等於電腦」。這裡指的不是硬體，不是作業系統，而是以應用為導向，以消費者為核心的思維。

因有這樣的理念，賣服務，賣應用，讓人更方便，也更加地讓眾人來開發這些應用軟體。

## 第二章 相關研究應用探討

先撇開程式設計，製作出一個應用軟體不講，我們要先探討的是，我們的定位在哪裡，如何來行銷，如何來獲利，以下列出幾點。

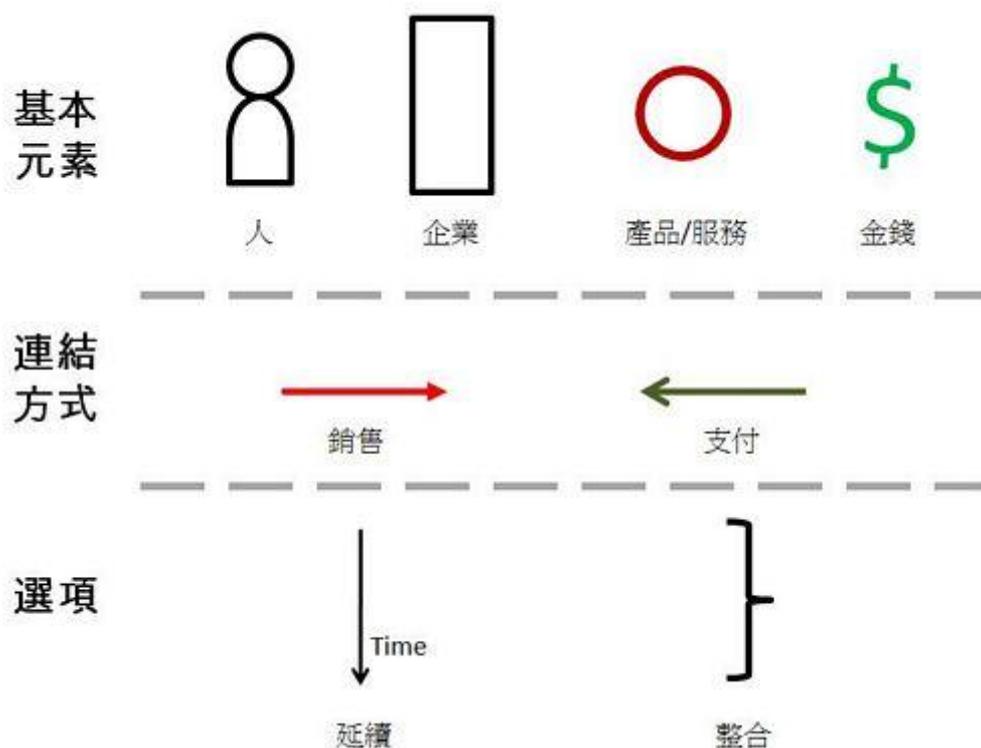


圖 1 基本元素

### 基本元素

要了解商業模式，用圖解的方式就可以清楚一目了然。而在繪製圖解時，我們需要先把商業模式中包含的各種元素先定義出來。其中會包含企業、個人、支付金錢、提供產品或服務、時間等。而運用這些基本元素，就可以組合變化出各式各樣的商業模式。

接下來，就要向各位介紹各種在App經濟中可以發展的商業模式。在以下的圖解中，標成綠色的「企業」指的就是App的發展商（開發者）

### 模式一：單純出售模式

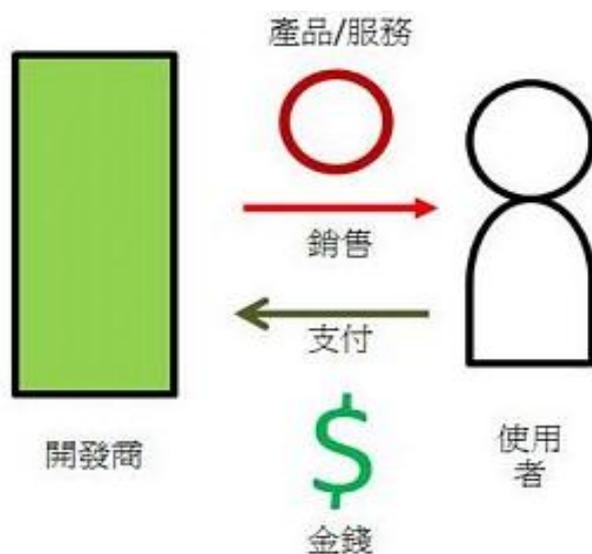


圖 2 模式一

圖2：使用者支付金錢購買App，開發者因而獲利

這種應該就是最單純的模式—開發者製作App，透過App Store或Market銷售給使用者。在這種模式中，重點是讓「單價」×「銷售量」所得的「銷售額」極大化，看起來似乎像是廢話，但是值得思考的是一假如某個App對特定族群來說是很有用的，但對於大眾來說也許不具吸引力，那麼與其定\$0.99但是也不會因此多賣幾個，是否反而應該把價格定高一些，然後透過正確的宣傳方式去讓有需要的人得知此資訊，雖然銷售量有限，但是因為單價夠高，整體銷售額也許更有利。且因為單價高，之後還有打折促銷的空間，進一步吸引價格敏感的消費者搶便宜。

## 模式二：廣告模式

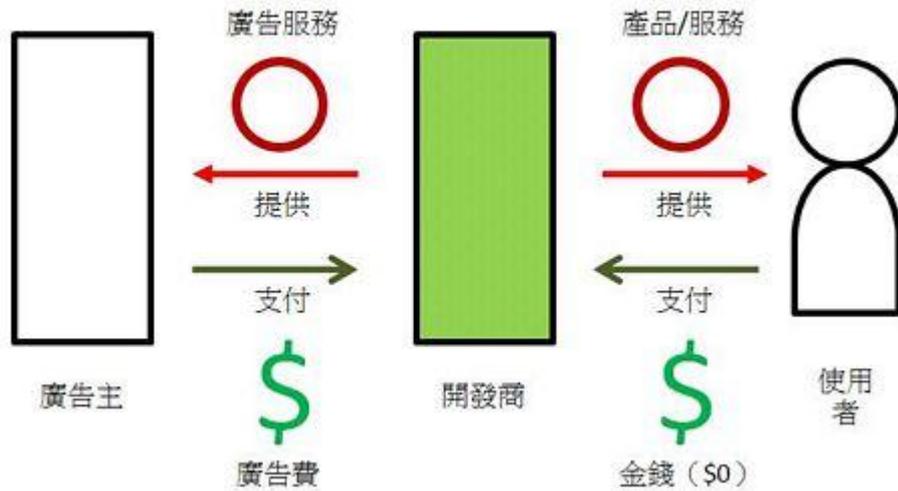


圖 3 模式二

圖3：使用者無需付費，廣告主支付廣告費給開發者，開發者因而獲利。

這也是相對單純的模式，這張圖在模式上有所簡化，實際運作在廣告主與開發者之間應該還有Apple或Google這兩大「廣告代理投放平台」才對。而此一模式的獲利主要就是靠廣告，因此要盡可能衝高App下載量，所以如果可以結合「使用者有需要」的服務例如資訊或情報，一來需求已經存在，二來廣告媒合效果也會更明顯。

### 模式三：收入組合模式

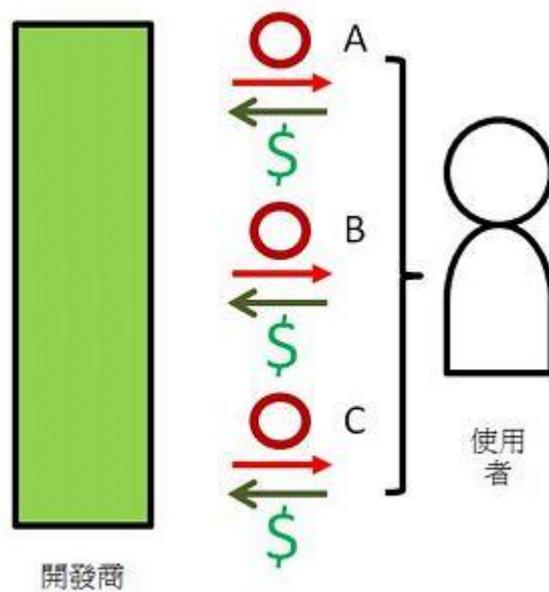


圖 4 模式三

圖4：使用者付費購買開發者所開發之多款App這是「單純出售模式」的延伸，意思是藉由其中一兩項特別便宜的產品吸引消費者上門，再順勢同時賣其他的產品給他，就像是現實生活大賣場通常會有所謂的DM商品吸引消費者到賣場消費，同時有機會購買其他的東西提高營業額是相同的道理。而在App的領域，甚至價格可以是\$0（搭配「廣告模式」一起運用）

#### 模式四：持續推出更新附屬功能模式

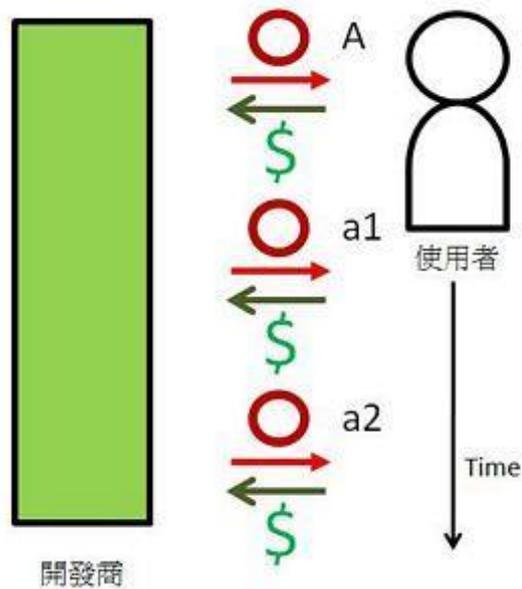


圖 5 模式四

圖5：使用者除了下載App主程式外，之後仍付費下載陸續推出的附屬功能。這裡指的是除了主程式之外，持續推出可以額外付費下載的附屬功能像是遊戲的新場景或是拍照軟體的新濾鏡效果等，讓收入可以持續增長。同樣的也可以讓主程式的費用是\$0（搭配廣告模式），或是運用「收入組合模式」的心理效果。

## 模式五：月租費模式

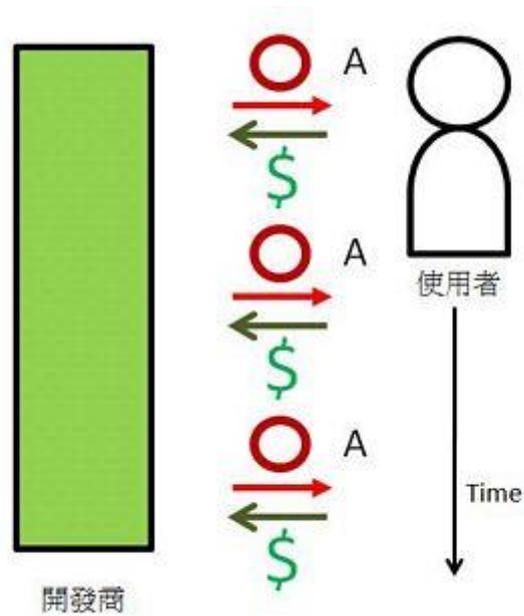


圖 6 模式五

圖6：使用者在持續使用App時定期支付金額給開發者顧名思義，只要使用者持續使用，隨著時間流逝就要定期付出費用。這種模式的變形也許綁的不一定是時間，而是使用量。而「訂閱內容」也會是可以運用此一模式的方式。

## 模式六：二次運用模式

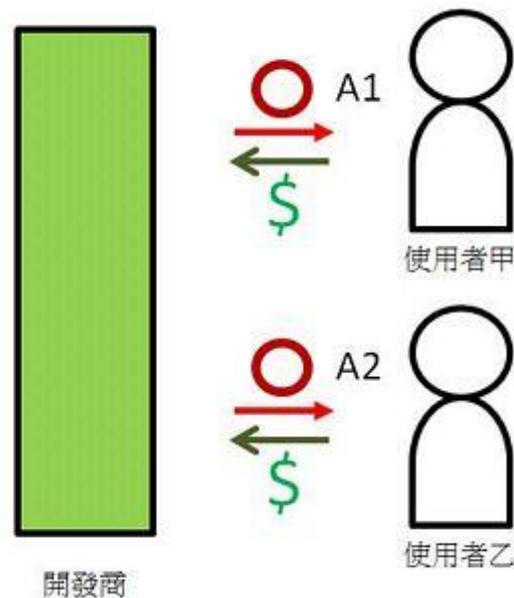


圖 7 模式六

圖7：開發者開發的App產品A銷售給某些使用者之後，重新組合或修改為產品B之後又可以賣給另一批使用者同樣的產品或服務，有沒有可能經過重新組合或篩選之後，變成另一款產品？真實生活中最典型的例子就是漫畫雜誌定期會把其中的連載集中再推出單行本，因此賣雜誌時賺一次，賣單行本時又賺一次。而這種模式在App領域除了「內容」之外，也可以成為「持續推出更新附屬功能」模式的變形。例如遊戲在持續推出新場景（「持續推出更新附屬功能」模式）後，隔一段時間再把其中比較受歡迎的場景組合之後推一個特價包，藉以吸引尚未下載的另一批顧客。

模式七：平台媒合模式

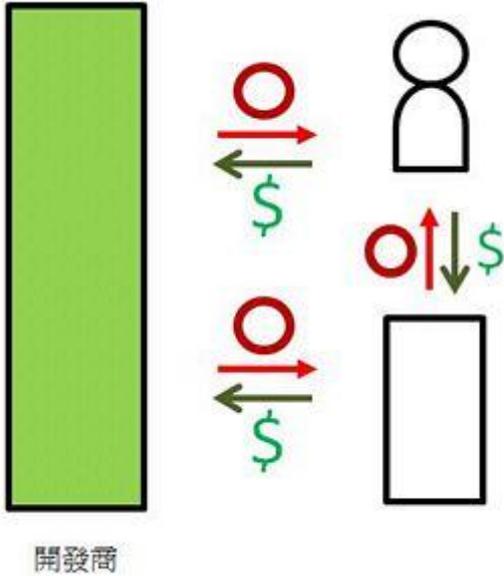


圖 8 模式七

圖8：開發者的App提供的是媒合使用者與企業，而除了App的功能之外，也可以借用其他網站的API甚至user graph舉例，像是LBS服務就可以發展出這類模式—媒合在地商家與正好人身處當地的消費者。對於商家來說，「平台媒合模式」比起單純的投放廣告也許更加精準，因此甚至願意投資更多的預算。

模式八：代為開發模式

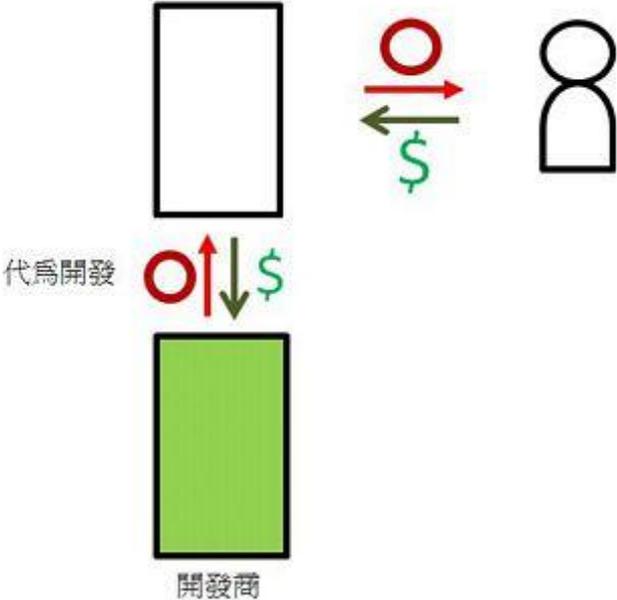


圖 9 模式八

圖9：開發者純粹做代工開發，幫有需要的企業製作這也是相當單純的模式，就是幫有需要的企業代為開發App，固然也許營收有限，但是不失為練功或是維持組織固定營收的方式。甚至有機會與上述的模式四、模式五、模式六混搭運用。

模式九：授權模式

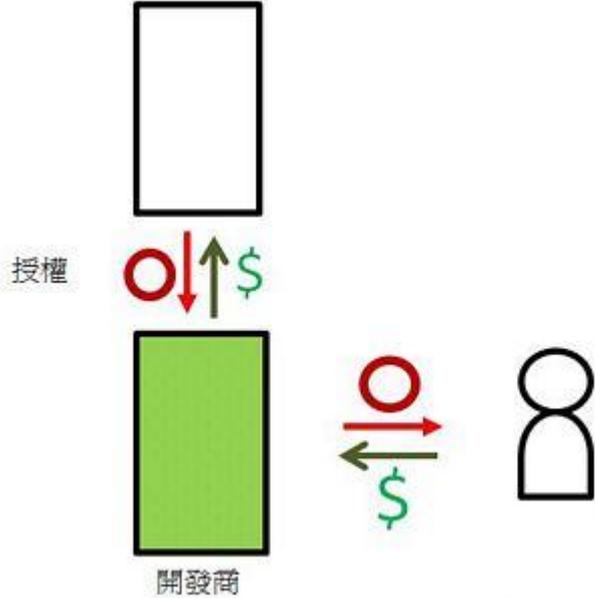


圖 10 模式九

圖10：開發者使用由其他內容供應者授權，使用其內容作為APP的內容  
內容與資訊不一定要由開發者自己產生或者只能使用 open 的內容，也可以以取得授權的方式與其他企業合作，同樣的，此一模式也可以與上述的模式四、模式五、模式六混搭運用。

### 第三章 系統原理與分析

#### 第一節 系統架構與系統流程

APP 與開發者與使用者的關聯



圖 11 APP 與開發者關聯

圖 11：我們藉由出版社的例子來看，作者可以把作品放在 APP 或者出版社的雲端系統，由此種途徑來銷售。

## 第二節 系統環境

### iPhone 作業系統介紹

iPhone OS(作業系統)包含了讓你的原生應用軟體能在iPhone及iPod Touch上順利運作的作業系統及技術。雖然他由MAC OS X作業系統上繼承了許多一般性的概念與很多基礎的技術，雖然使用者的需求有些許的不同，但iPhone的作業系統仍被設計為最適合行動裝置的使用。若你已經是MAC OS X作業系統的開發者，則你會發現這兩者有很多熟悉的技術，但你仍會發現，也有不少技術是僅適用於iPhone作業系統的架構，如多重觸控的介面及加速計的支援等等。



圖 12 iphone 產品圖

iPhone 的SDK軟體發展套件包含了程式碼、開發資訊及你在開發過程中所需要的程式撰寫、軟體測試、軟體執行、程式除錯以及軟體最佳化調整等工具。而Xcode工具為了支援 iPhone作業系統上的開發，因此也做了版本上的更新。除此之外，為了提供一個你程式碼基本的編輯、編譯與除錯的開發環境，Xcode也提供了一個進入點(launching point)讓你在iPhone與iPod Touch上測試你的程式，Xcode甚至允許你在iPhone的模擬器上執行你所撰寫的應用程式。

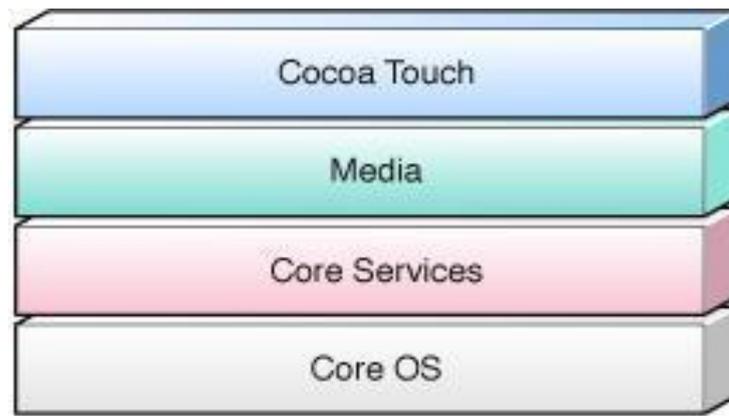


圖 13 iPhone 作業系統應用技術層

在iPhone 的作業系統中，基礎的系統架構以及許多的技術應用都和Mac OS X是相同的。而iPhone作業系統的核心也是基於Mac OS X內各式不同的Mach核心。在此核心的最上面一層的服務是用來創建此平台上的應用，圖1展示了這些不同技術層的應用描述。

這些分層檔你在撰寫你自己的程式碼時提供了不同的應用選擇。舉例來說，Core OS (作業系統核心)以及 Core Services(服務核心)層包含了iPhone 作業系統最基礎的介面，如檔案的存取、低階的資料型態、系統服務及網路通信協定等等。

這些的介面大多以C語言為基礎並包含了Core Foundation、CFNetwork、SQLite、存取POSIX序列以及UNIX協定等等的技術。

而當你進入更高的一層，你會發現更進階的技術都是以C以及Objective-C 做為基礎的介面。舉例來說，Media layer包含了2D與3D的繪圖支援及音效與影片的處理。這一層包括了同樣以C為基礎技術的OpenGL ES、Quartz 及Core Audio。同時並包含了也是以進階Objective-C為基礎的動畫引擎Core Animation。

而在 Cocoa Touch 這一層，大多數的技術是使用Objective-C。而這一層則提供了一個建構程式的最基礎架構。舉例來說，Foundation framework 提供了一個對集合、檔案管理、網路操作及其他物件導向應用的支援。而UIKit則提供了一個應用程式在視覺化的基礎架構，包含了視窗的classes、views、controls控制元件以及管理這些物件的控制器等等。在此層的其他frameworks(框架)則提供你存取iPhone內聯絡人及照片 資訊，以及加速計狀態及其他硬體特徵等等。

對很多新的專案來說，Cocoa Touch layer是一個開始的起點，特別是UIKit framework。當你決定是否要使用其他的技術時，請由高階的framework開始，而當有需要的時候再來研究低階的framework。因為高階的framework能讓你用最少的力氣去達到支援標準的系統行為。而唯一你會去接觸低階的framework則是當你發現這些高階的framework 都無法提供你的需求時，此時你才必須利用更低階的技術來處理這些特別的動作或功能。

iPhone SDK支援建立一個圖形化介面的iPhone原生應用程式，而你的應用程式將與iPhone內建的Photos照片、Weather天氣以及Clock時 鐘等系統應用程式一樣，都是在iPhone的主畫面上可以找到的。當你的程式被執行的時候，除了iPhone系統核心及一些低階的服務之外，你的軟體就是 系統唯一運作中的程式，此時，你的程式將佔滿整個iPhone的螢幕，而使用者的注意力也將停留在你程式所提供的畫面與功能上，而當使用者按下Home鍵 的時候，你的程式會自動結束並再一次的回到iPhone的主畫面。讓程式獨佔系統是有優勢的，因為他能提供對系統資源的最完整控制，此外，你還能搭配你的 程式碼去充份利用iPhone內建的加速計、攝影機以及繪圖卡等硬體。

使用者與iPhone或iPod Touch裝置溝通的方式基本上和Mac OS X有很大的不同，當然，在程式的設計上也是不太一樣。

在iPhone 的程式中，沒有所謂的多重或分割視窗的概念，取而代之的是以單一視窗來顯示所有的內容，這能讓你用一個全新的views與controls來建立你的應用程式，並以一個組織性的管理來呈現你所提供的資料。除此之外，很多標準的views與 controls，他的運作模式也和Mac OS X上的使用有些許的不同。而這大多數的改變是很明顯的，但有一些仍需要去重新思考一個新的方式來組織與呈現你的資料。

在iPhone 作業系統中的事件驅動模型(event-handling model)，和傳統的桌面型應用軟體來比較，也代表著一個相當重要而有意義的開始。取代一般完全依靠滑鼠與鍵盤的方式，iPhone 作業系統展示了一個觸控的全新概念。一個觸控的訊息在任何時間都能發生並且還能結合其他單一或多個額外的觸控事件。這些觸控的訊息能偵測到使用者與你內容 所做的簡單互動動作，如選取或拖拉；或是複雜的手勢或互動如敲擊(swipes)、或捏放(pinch-open)及捏擠(pinch-close)等手 勢動作(舉例來說，就像在iPhoto照片程式裡對照片所做的放大、縮小等動作)。

在思考你應用程式的基本架構之前，你必須先想到你的使用者將如何使用你的程式，iPhone的軟體應該是簡潔並專注在使用者當下的需要。請記住，對想要隨時能以最快的方式來取得資訊的使用者來說，她們並不想浪費太多時間在你程式的各個畫面中尋找功能或答案。用最簡單的方式來強調使用者

最需要的主要資訊是相當重要的。而對遊戲或其他趣味性的程式來說，你必須想到你的使用者將如何的與你的程式互動，並在適當的地方藉由iPhone內建的加速計及攝影機等技術來產生樂趣。

而當你準備開始進行程式開發的時候，你必須先了解的是Foundation及UIKit這兩組framework。這些frameworks提供了所有iPhone程式所需要的主要服務與功能，而你也必須研究系統中其他的frameworks以了解這些frameworks還能提供哪些其他的服務。這些framework中的文件都包含了連結及相關必要的資訊來幫助你學習如何進階的使用這些frameworks。

從什麼地方開始利用Xcode內建提供的預設專案範本來建立功能單純的應用程式是相對比較簡單的，但若是寫一支真正有用而且界面又漂亮的軟體或程式，那你勢必得花一些時間來閱讀這些可輕易取得的線上文件及手冊。

至少，你必須花一點點時間來閱讀在iPhone Dev Center中關於tools工具及建立程式的相關文章。如果你對iPhone是Mac OS X作業系統一點也不熟悉，那你得花點時間去了解這些與iPhone作業系統相關的基本設計方法及規定，這對撰寫iPhone平台的程式是相當重要的。在iPhone的作業系統中，若能善用這些frameworks所提供的超強功能，你就能讓程式跑的有效率而且最佳化。但如果你花時間只學這些基本frameworks的功能則你會發現建立你的應用程式是比只學習這些基本且使用中的設計方法還要來的困難。每一個iPhone的開發者都必須要知道的基本資訊就是Cocoa Fundamentals Guide。這一份文件涵蓋了Objective-C語言的基礎、程式設計的規則及設計的方法，

而這都常被使用在UIKit及其他系統的frameworks。一但你了解了這些基本的規則並開始建立你的iPhone程式之後，你可以開始參考iPhoneOSProgramming Guide 讓你在開發過程中取得特定的功能細節。這本書提供了開發iPhone程式的主要概念及以任務為基礎的資訊，包括了如何顯示使用者介面，控制系統訊息及事件等範例程式，並能夠確實掌握住iPhone作業系統的特色與能力。同時，這本書還描述了整個開發過程並對如何使用Xcode來進程式的開發有相當詳細的介紹。

### 第三節 使用軟體：



圖 14 MAC OS X

### 作業系統：MAC OS X

雖然目前最新的 Microsoft WINDOWS 作業系統版本是 Windows 7。但這次我們專題使用的是 MAX OS X 作業系統，因為要寫蘋果的程式，只能用蘋果系統。

OS X(前稱*Mac OS X*)是蘋果公司為麥金塔電腦開發的專屬作業系統Mac OS 的最新版本。Mac OS X於2001年首次在市場上推出，並從2002年起隨麥金塔電腦發售。它是一套Unix基礎的作業系統，包含兩個主要的部份：核心名為Darwin，是以FreeBSD原始碼和Mach微核心為基礎，由蘋果公司和獨立開發者社群協力開發；及一個由蘋果電腦開發，名為Aqua之專有版權的圖形使用者介面。

OS X Server 亦同時於 2001 年發售，架構上來說與工作站（客戶端）版本相同，只有在包含的工作群組管理和軟體工具上有所差異，提供對於關鍵網路服務的簡化存取，像是郵件傳輸伺服器，Samba 軟體，LDAP 目錄伺服器，以及名稱伺服器（DNS）。同時它也有不同的授權型態。

### 顯著特點

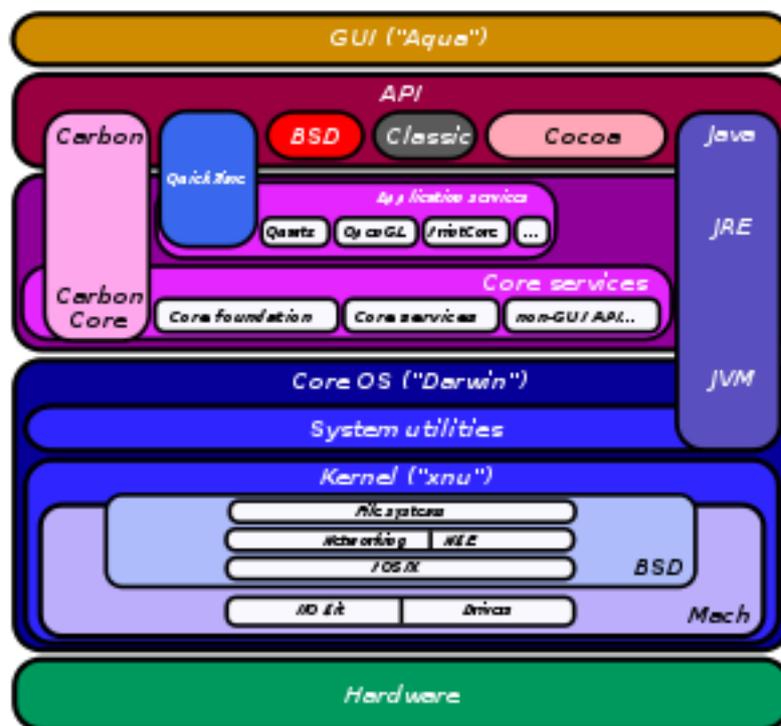


圖 15 OS X 架構圖

- 全64位元技術，可以通過第三方的Mode Selector進行64位元和32位元切換。
- Grand Central Dispatch. 幫助程序編譯人員使用多核心編譯，增強軟體多核心支持。
- OpenCL。是一種讓圖像處理器發揮極致來加速應用程序性能的新技術
- 輔助功能。通過萬能輔助功能包含的VoiceOver 2可以讀出任何Mac上的文字。Multi-touch觸控版支持中文手寫。含有40種以上盲文現實，支持藍牙模式。
- 它的Quartz影像模式使用PDF（可攜文件格式，Portable Document Format）的子集合當作基礎。
- 全彩，可連續縮放的小圖示（最大到 128×128（10.5 Leopard 最大到 512×512）像素）
- 在視窗周圍的陰影和分離的文字元素來提供深度的感覺。
- 依靠NeXT-型程序服務，提供全局拼寫檢查及其他功能強大的工具
- 專用界面工具集、文字、圖形和視窗元件的反鋸齒效果
- 新的界面元素，包括「紙單」sheets（文檔模型對話框附帶於特殊窗口）「抽屜」drawers.
- Interweaving windows of different applications (not necessarily adjacent in the visible stacking order).
- 色彩同步在核心繪圖引擎中內置色彩匹配（為印刷和多媒體專業人士）。
- OpenGL（在10.2版中納入）合成視窗到螢幕上，容許硬體加速繪圖。這種技術稱為Quartz Extreme。
- Exposé（在10.3版中納入）可以快速地排列視窗或是顯現桌面。在10.6中進行大量改良，直接從Dock上激活Exposé

- 在整個作業系統中普遍使用Unicode。
- Spotlight搜尋技術（在10.4版中納入）允許以項目的特性或內容來快速的即時搜尋資料檔案、郵件訊息、照片、和其他資訊。
- Automator（10.4版中加入）是一種設計來為不同的工作任務建立一種自動化的工作流程之應用程式。

Xcode前身是繼承自NeXT的Project Builder。

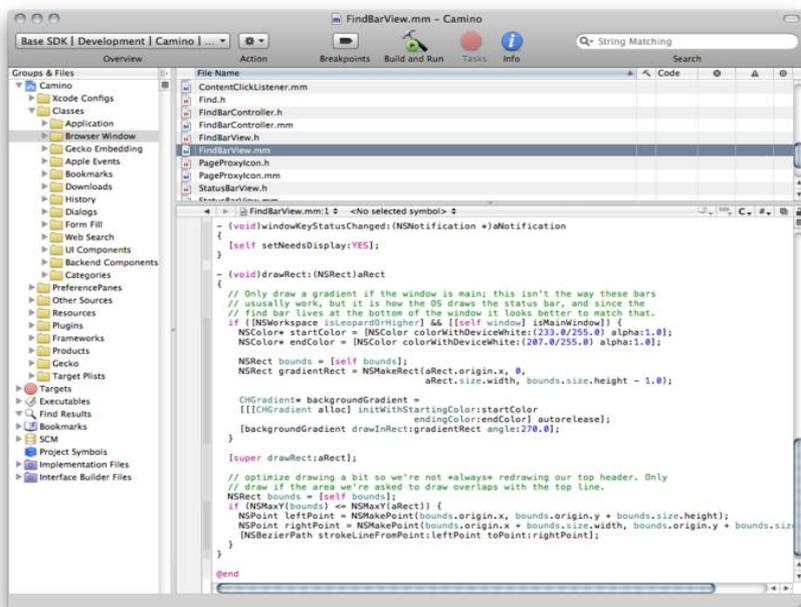


圖 16 Xcode

The Xcode suite 包含有GNU Compiler Collection自由軟體（GCC、apple-darwin9-gcc-4.0.1 以及 apple-darwin9-gcc-4.2.1, 默認的是後者），並支援 C語言、C++、Fortran、Objective-C、Objective-C++、Java、AppleScript、Python以及Ruby，還提供Cocoa、Carbon以及Java等編程模式。協力廠商更提供了 GNU Pascal，[1] Free Pascal[2], Ada[3], C Sharp[4], Perl[5], Haskell[6] 和 D語言。Xcode套件使用 GDB作為其後台調試工具。從Xcode 3.1開始附帶iOS SDK, 作為iOS的開發環境。

Xcode最新主要版本是Xcode 4.0，已於2011年3月9日正式發行。從本版本開始，非Apple Developer註冊會員亦能從Mac App Store中付費下載，收取US\$4.99的費用。

從 Xcode 4.1 開始，針對 Mac OS X v10.6 及 Mac OS X v10.7 使用者從 Mac App Store 免費下載

## 影像編輯軟體：Photoshop



圖 17 Photoshop

Photoshop 是專業的美工繪圖軟體，透過軟體實際的影像編修技巧，會使你工作上的任務更有效率。它具有完備且功能強大的工具指令；不論是影像的編修處理、或是繪製合成，它都有細分的各種工具及相關的數值設定面板，可以做到最精準的調整；而強大的濾鏡功能更是不能錯過的一項，各種濾鏡特效可以模擬出以假亂真的影像畫面，其鬼斧神工的效果令人驚嘆。除此之外，Photoshop 的向量式的繪圖功能，及更多網頁元件的製作工具，使它成

為一套能橫跨平面、多媒體和網頁製作的影像處理軟體。因此，對設計者來說，Photoshop 將會是最佳的專業影像處理軟體。

Photoshop 可說是平面媒體在影像至製作上的首選之一，但先前的版本更新主要還是鎖定在影像處理的功能上以及校色功能上的更新，不過現在 Adobe 公司也注意到網路這塊市場大餅，於是將先前 Adobe 所推出的網頁影像處理軟體 Image Ready 直接整合到 Photoshop 之中，讓 Photoshop 在網頁影像的處理上更可以有水準之上的表現。

## 第四節 系統特色

本次專題以圖形點擊及英文發聲為主要設計架構，讓使用者能藉由手機的方式，讓每個人能隨時的練習英文，讓瀏覽者能夠隨時隨地不受空間、區域的限制，大大提高學習的便利性，對瀏覽者而言，有以下幾個吸引成為消費者的優點：

1. 提供一個快速簡便的小孩教學管道
2. 24 小時隨時都能消費購買
3. 提供英語的階級更新（目前為針對小孩學英語）

對手機如此普及的世界裡，不但可提高學習意願，且不受地區限制。不過需要提到的是，這次專題製作的軟體目前還在初階階段，並無法做到更新或附加購買的配套程式…等，而在現今大多數的程式都可以採用 A P P 更新或者介面上加值程式 …等的問題，所以這些部份我們並沒有實際製作出這些功能。

## 第五節 進度規劃

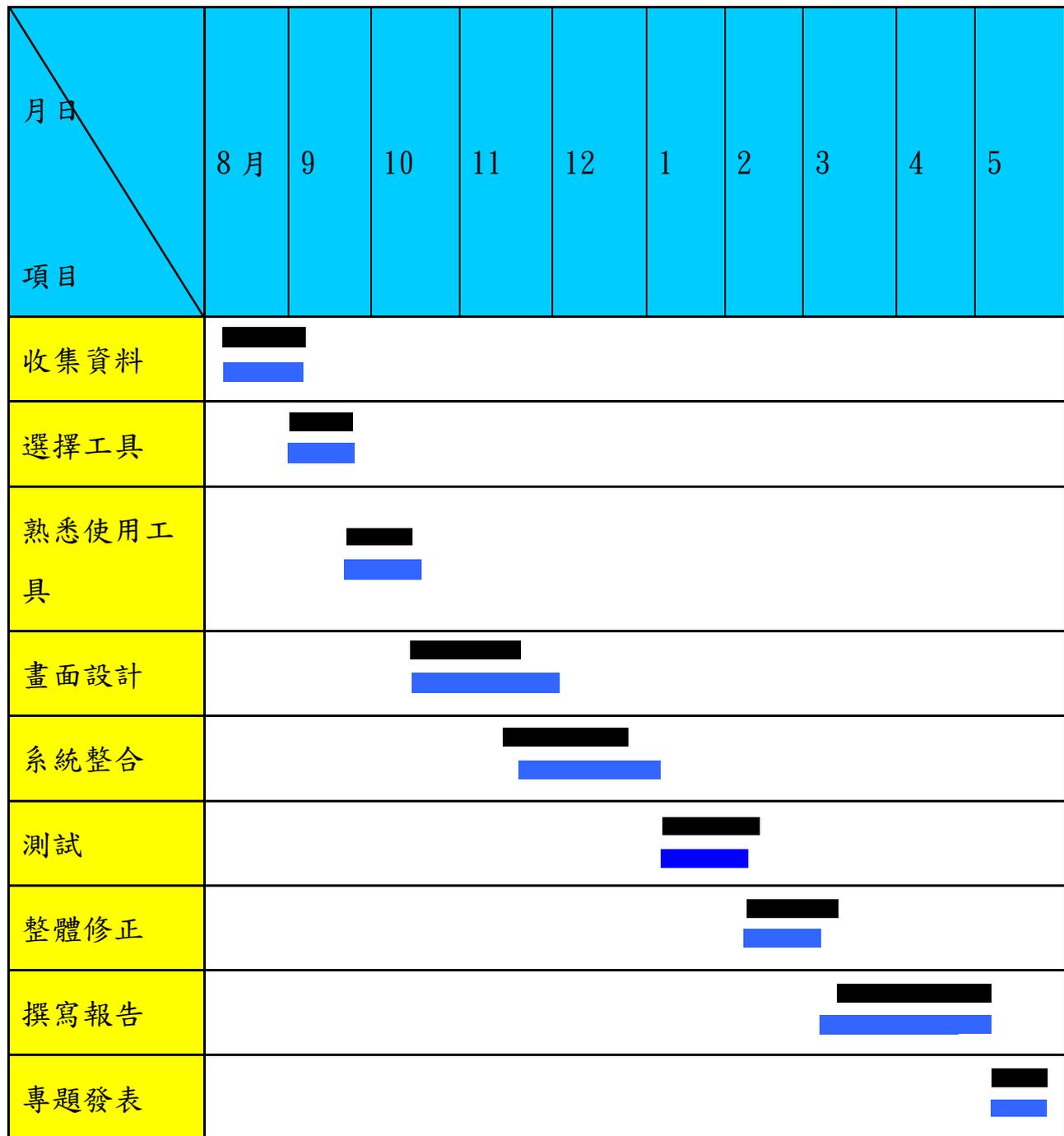


表 1 甘特圖 (Gantt Chart)

■ 計劃      ■ 實際

## 第六節 人力分配

項目/組員	劉俊麟	王碩志	樂愷	張志豪	江俊杰
系統規劃	★		★	★	★
系統分析	★		★		★
系統設計	★		★		★
資料蒐集	★		★		★
程式製作		★		★	
影像處理		★		★	
後端製作		★	★	★	
軟體應用		★		★	
程式設計		★		★	
程式修改		★		★	
系統測試	★		★		★
系統建置	★	★			★
報告撰寫	★		★		★

表 2 工作分配表

## 第三章 專題實作

這次的專題，是以兒童初學英文字母及吸引孩童的學習意願為架構，以三項要素為特色重點：

### 1. 簡單易懂

整個視窗一目瞭然，隨按隨出聲。

### 2. 精美圖片

各個英文字母均有吸引小孩兒美化的字母

### 3. 架構完整

設計出一個架構完整，使用迅速方便，包含點擊圖片，與發聲練習。

本專題的實作過程中，我們一步步的希望以上述三點為基礎來製作一個好的

兒童學習字母APP，並經過一系列探討來確認製作方式：

- (1). 確定系統方向：先確定此系統的需求、目的、方向，以此為目標設計。
- (2). 系統規劃：根據系統需求來做系統的功能。
- (3). 系統分析：討論研究系統的功能並加以分析。
- (4). 系統設計：依系統分析的結果來設計系統。
- (5). 檢驗系統：系統設計的內容是否滿足系統需求。
- (6). 程式的整合與測試：將所有的程式有規律的統整與測試程式是否執行順暢。
- (7). 系統評估維護：系統將完成時，系統開發者對系統做功能效率的評估；  
管理者定期維護系統。

## 第四章 系統操作

### 第一節 系統環境安裝

開始安裝的畫面，只有一個 繼續 的鈕可以選。工具列上也沒啥東西好點的。



圖 18 安裝畫面--1

軟體協議。理所當然，同意。



圖 19 安裝畫面---2

再次確定。

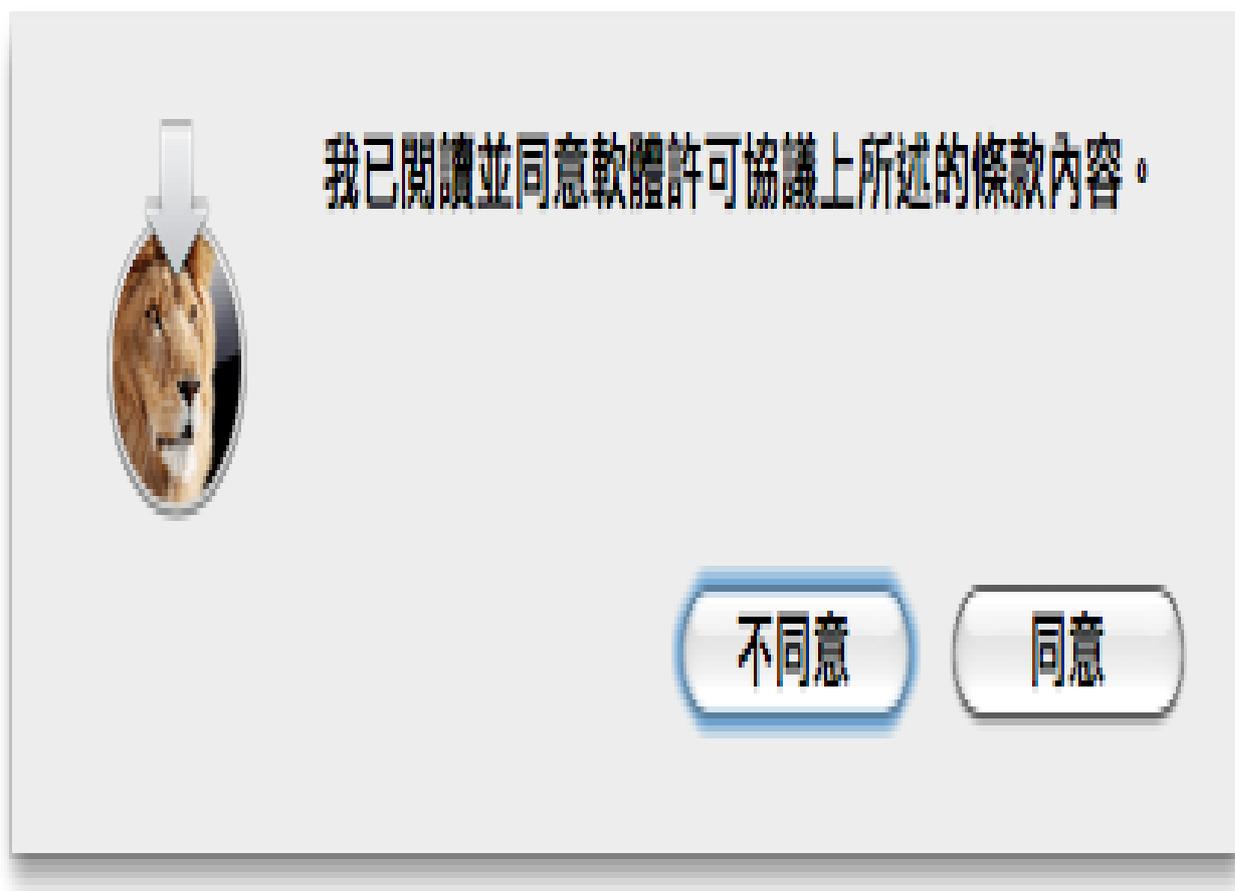


圖 20 安裝畫面--3

點選 顯示所有磁碟... 鈕來選擇安裝磁碟。



圖 21 安裝畫面--4

這裡比較要注意的是，一般大家都是從 10.6 來執行這個安裝程式的。此時就算你有連結上外接硬碟，且該硬碟處在正確的安裝格式，這裡仍舊只會顯示出內置的硬碟。因此想要保留原本的 10.6 並將 Lion 安裝在外置硬碟的朋友們，得另尋他法。



圖 22 安裝畫面--5

接著輸入使用者密碼來允許安裝。



圖 23 安裝畫面--6

程式開始準備安裝。這裡是安裝前最後一個可以反悔的機會。等系統準備完後，等等在看到桌面就就是 Lion 囉～



圖 24 安裝畫面--7

系統將會自動重新開機以開始安裝



圖 25 安裝畫面--8

若是有程式尚未關閉，就會出現下面這個畫面。



圖 26 安裝畫面--9

因為 尚未學會怎樣截圖，所以下面是用 iPhone 照的…

Apple 說 Lion 學習了相當多 iOS 的元素，就是所謂的 back to Mac

從這個安裝畫面就可以發現了。看看這灰色背景，有使用過 iPad 的人，是否覺得很熟悉。



圖 27 安裝畫面--10

真高興看到這個安裝成功的畫面，安裝的時間比預計的短。



圖 28 安裝畫面--11

系統自動重新開機後，可能是因為升級安裝的關係，經典的歡迎動畫並沒有出現

以前那些繁複的設定與註冊，都不用再弄一次，直接進入桌面。

桌面會出現一個 Mac OS X 設定輔助程式。因為 Lion 預設的捲視方式跟 iOS 相同，也就恰巧跟我們所習慣的電腦使用方式相反，因此這個程式的唯一目的，就是讓我們學會這樣的捲視方式。



圖 29 安裝畫面--12

使用正確的捲視方式將畫面拉到最底，就可以看到 開始使用 Mac OS X Lion 的按鈕了。



圖 30 安裝畫面--13



當安裝好 MAC 的虛擬平台後，接下來就可以開始來開發 iPhone 的應用程式，但打開 MAC 後一切空空如也，到底開發工具在那呢？哈~想要開發 iPhone 的應用程式，得先去 Apple 申請個開發帳號，並下載開發工具 Xcode，將子才可開發 iPhone 的應用程式，而申請開發帳號並不用付任何的費用，只要填填表單，註冊完就可下載囉!現在梅干就來介紹一下，如何申請與安裝 iPhone 的開發應用工具。

## Step1

開啟瀏覽器，於網址列輸入

<http://developer.apple.com/devcenter/ios/index.action> 進入後，點選  
右上角Register選項。

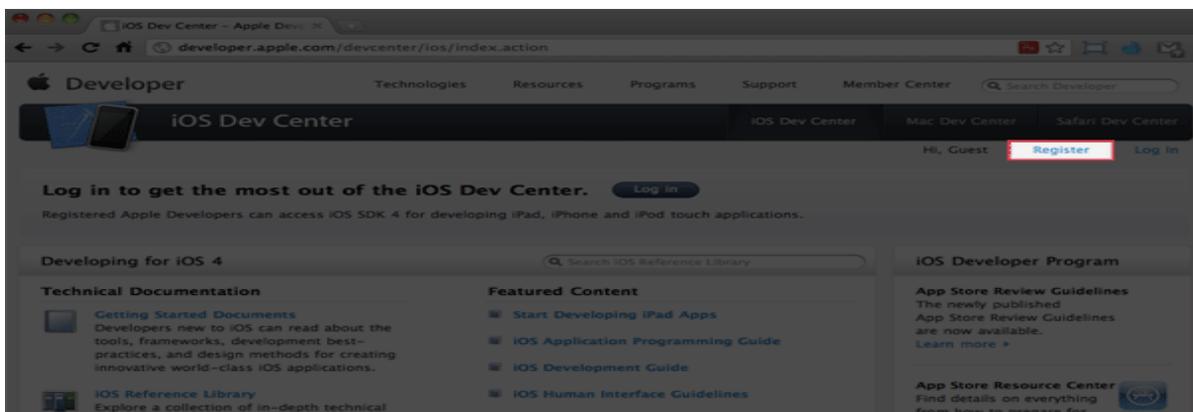


圖 31 Xcode 註冊畫面--1

## Step2

再點GetStarted按鈕。

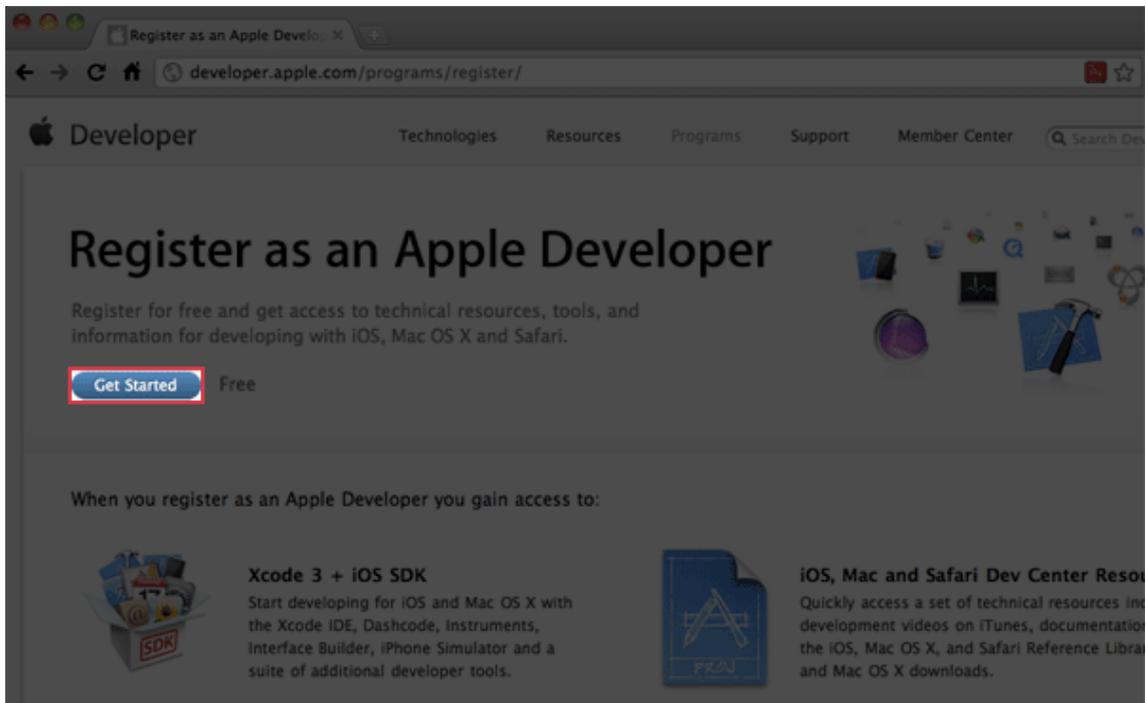


圖 32 Xcode 註冊畫面--2

## Step3

選擇Creat an Apple ID 選項

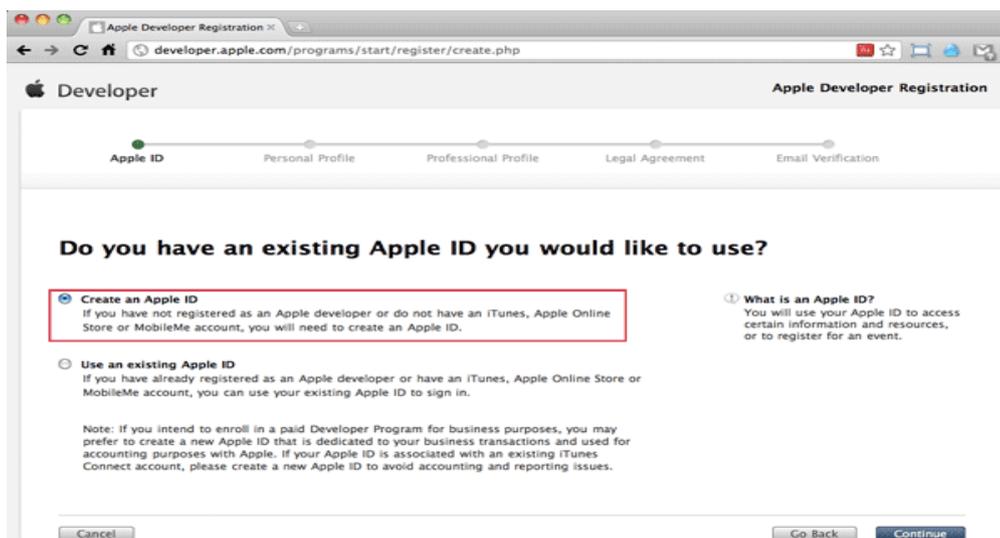
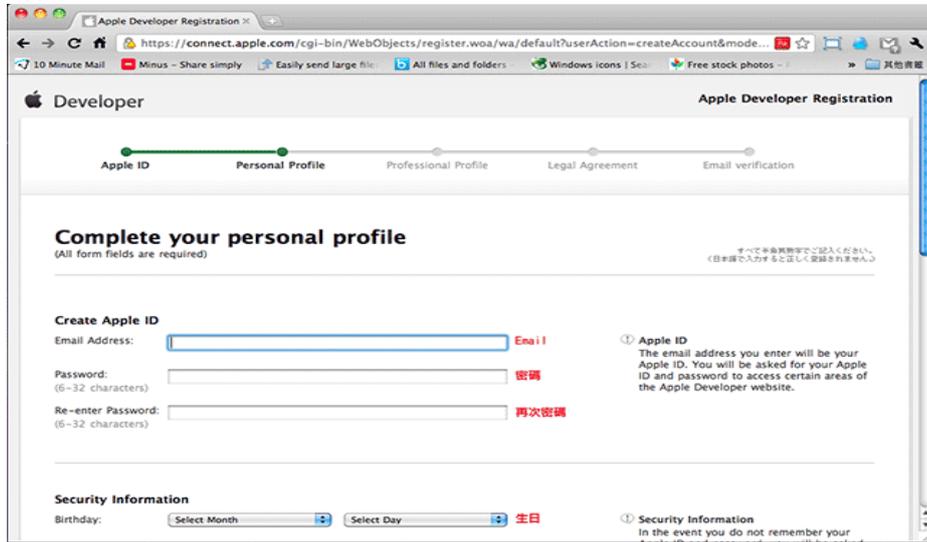


圖 33 Xcode註冊畫面---3

## Step4

接下來就依照欄位填寫，但Email一定要確實填寫，否則待回會收不到認證碼。

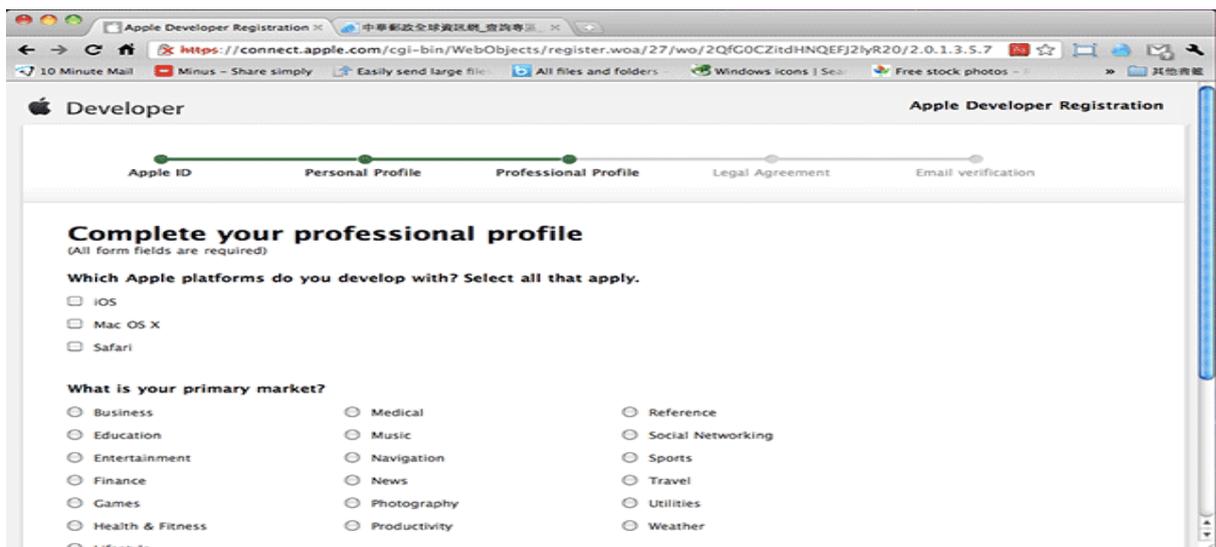


The screenshot shows the 'Complete your personal profile' step of the Apple Developer Registration process. The progress bar at the top indicates that 'Personal Profile' is the current step, with 'Apple ID' completed and 'Professional Profile', 'Legal Agreement', and 'Email verification' remaining. The main heading is 'Complete your personal profile' with a note '(All form fields are required)'. Below this, there are three input fields: 'Email Address' (with a red 'Email' label), 'Password' (with a red '密碼' label), and 'Re-enter Password' (with a red '再次密碼' label). To the right of these fields is a help icon and text: 'Apple ID The email address you enter will be your Apple ID. You will be asked for your Apple ID and password to access certain areas of the Apple Developer website.' Below the password fields is a 'Security Information' section with a 'Birthday' field consisting of 'Select Month' and 'Select Day' dropdowns, and a red '生日' label. To the right of this is another help icon and text: 'Security Information In the event you do not remember your'.

圖 34 Xcode 註冊畫面—4

## Step5

設定基本資料，眼睛閉起包隨便點一點就好了。



The screenshot shows the 'Complete your professional profile' step of the Apple Developer Registration process. The progress bar at the top indicates that 'Professional Profile' is the current step, with 'Apple ID' and 'Personal Profile' completed and 'Legal Agreement' and 'Email verification' remaining. The main heading is 'Complete your professional profile' with a note '(All form fields are required)'. Below this, there are two sections: 'Which Apple platforms do you develop with? Select all that apply.' with checkboxes for 'iOS', 'Mac OS X', and 'Safari'; and 'What is your primary market?' with radio buttons for various categories: Business, Education, Entertainment, Finance, Games, Health & Fitness, Lifestyle, Medical, Music, Navigation, News, Photography, Productivity, Reference, Social Networking, Sports, Travel, Utilities, and Weather.

圖 35 Xcode 註冊畫面--5

## Step6

沒得選擇，只能勾選。

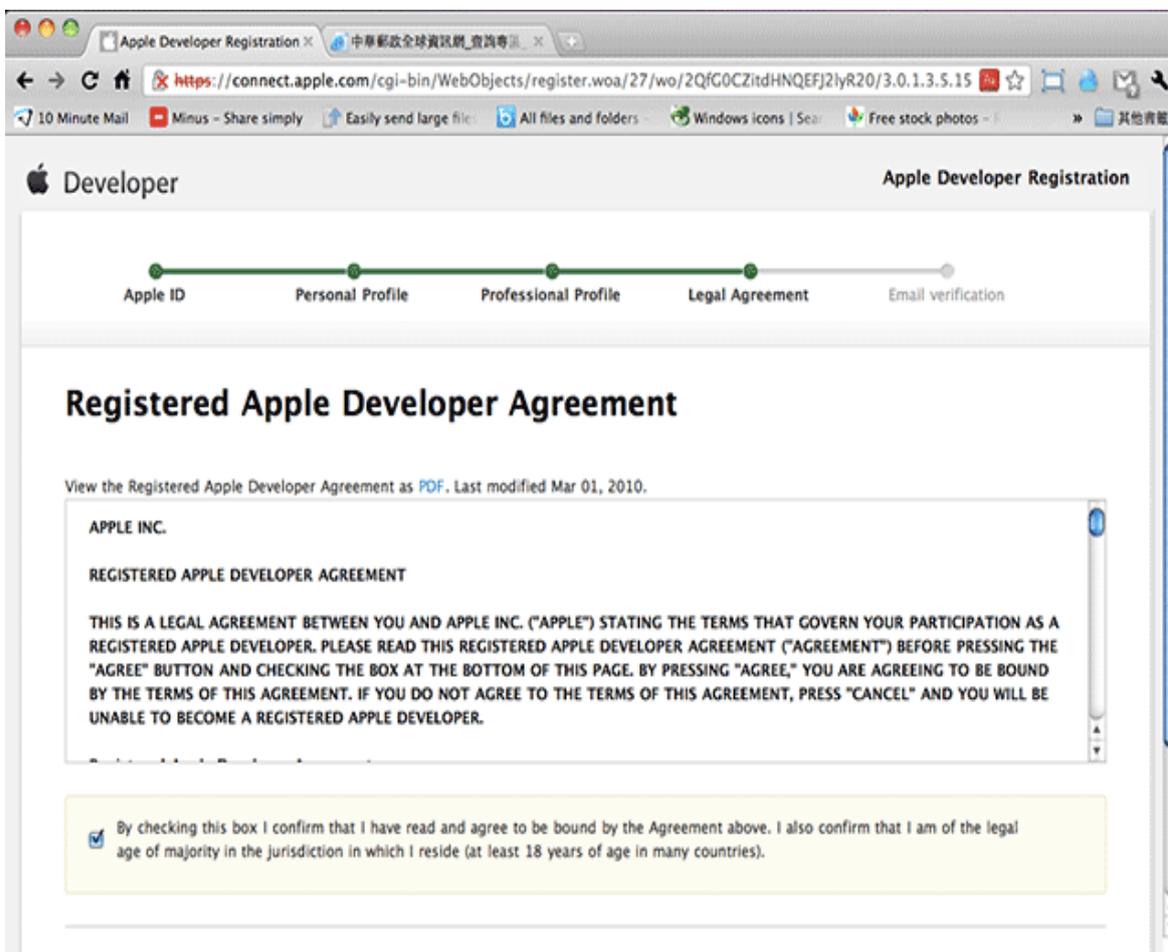


圖 36 Xcode 註冊畫面--6

## Step7

接下來收一下信箱，就會看到驗證碼。

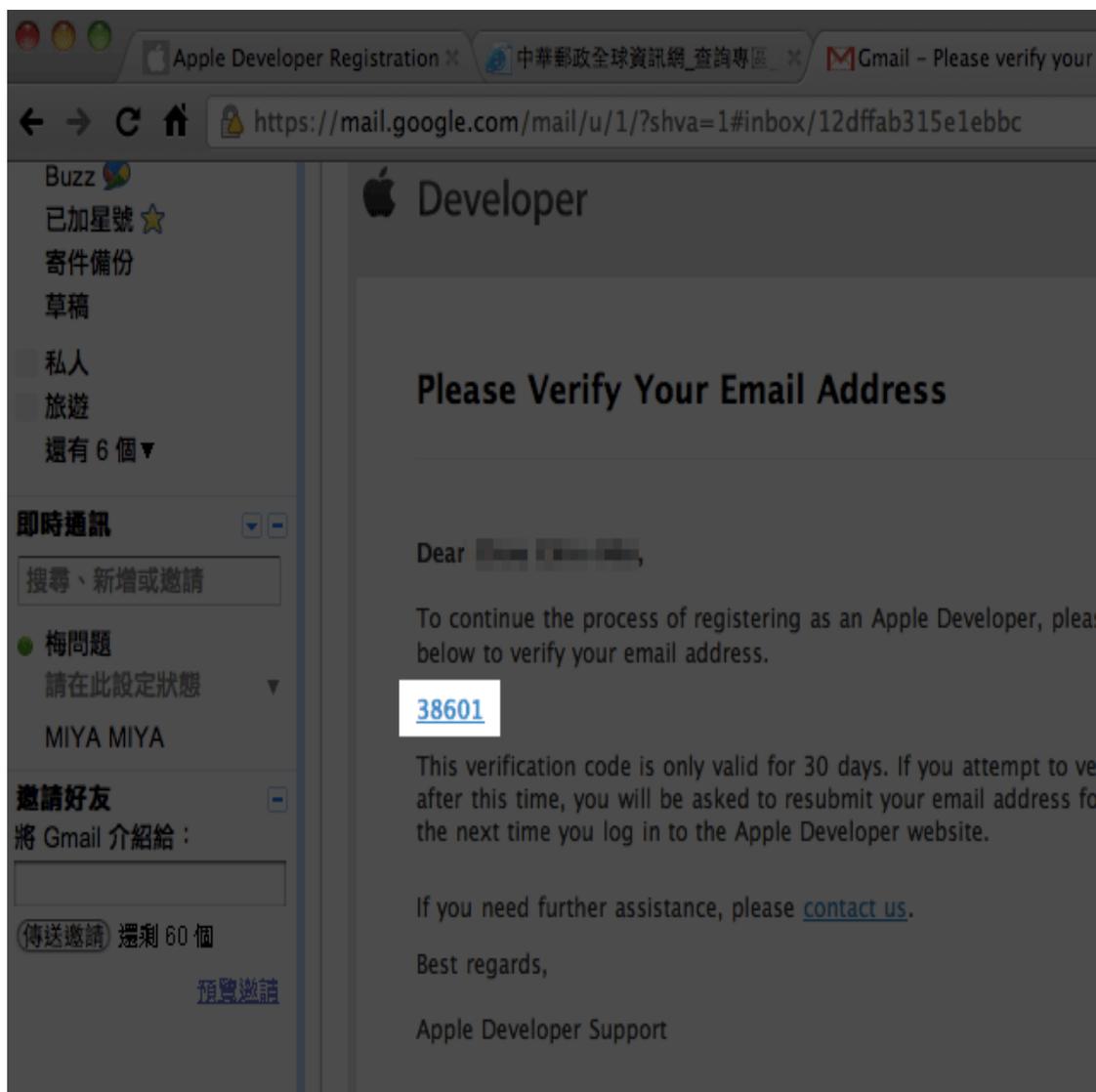


圖 37 Xcode 註冊畫面--7

## Step8

填入信箱中的驗證碼。

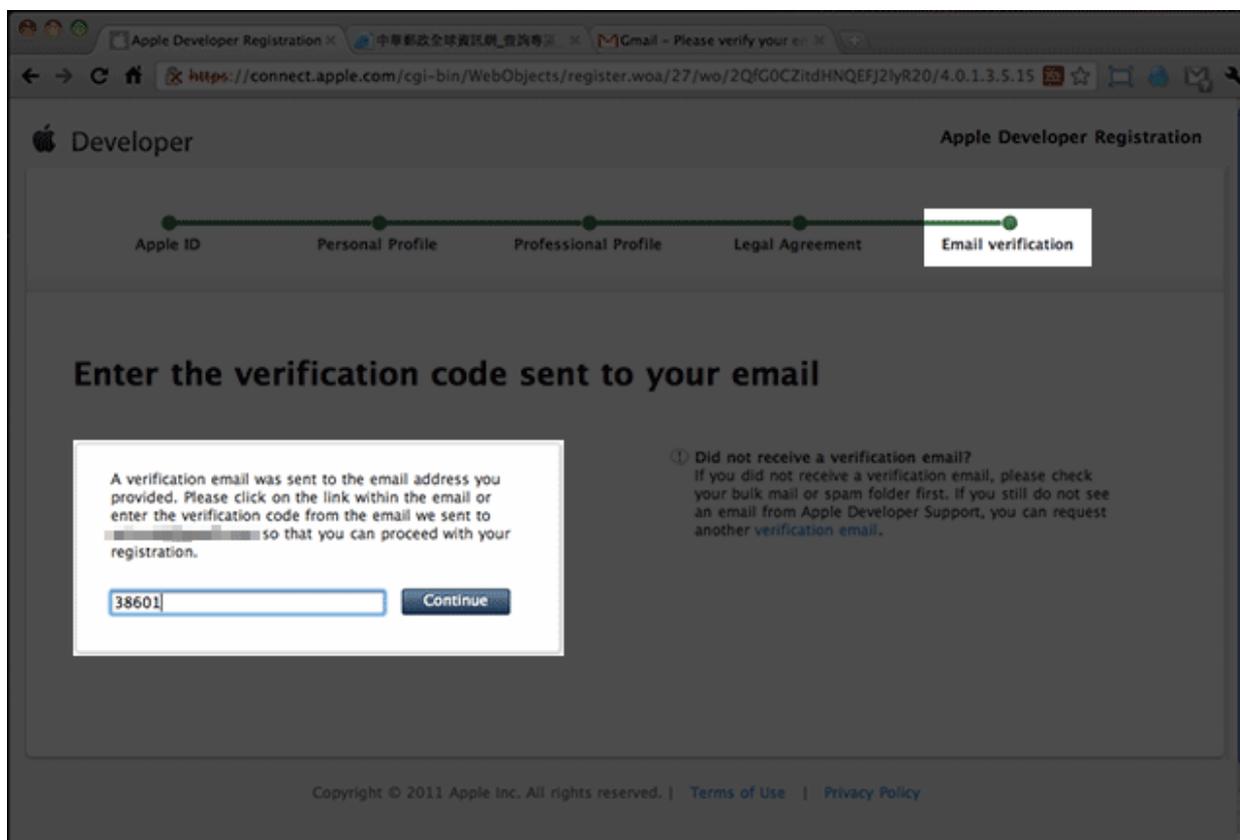


圖 38 Xcode 註冊畫面--8

## Step9

這樣就完成註冊囉!再按下Continue鈕。

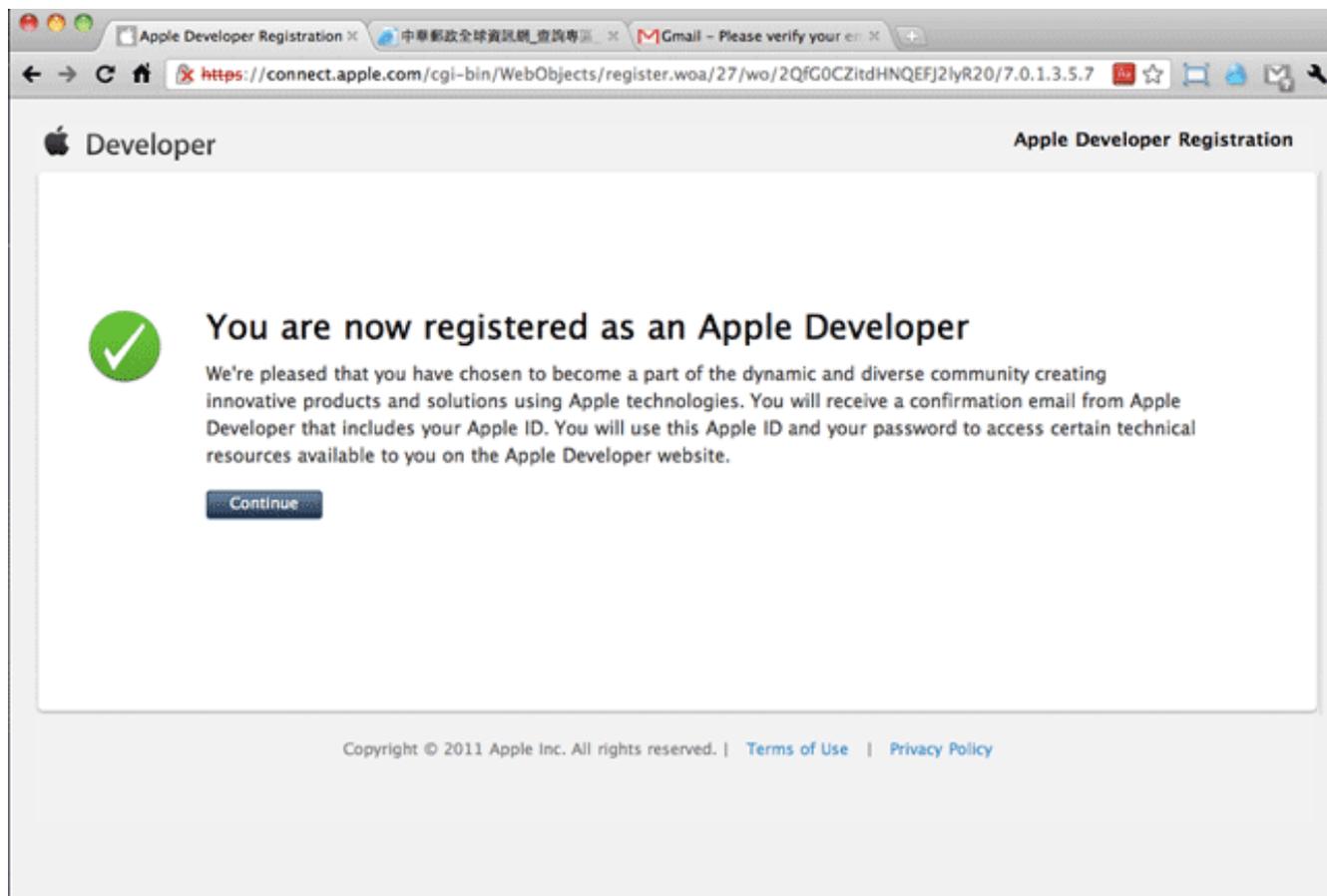
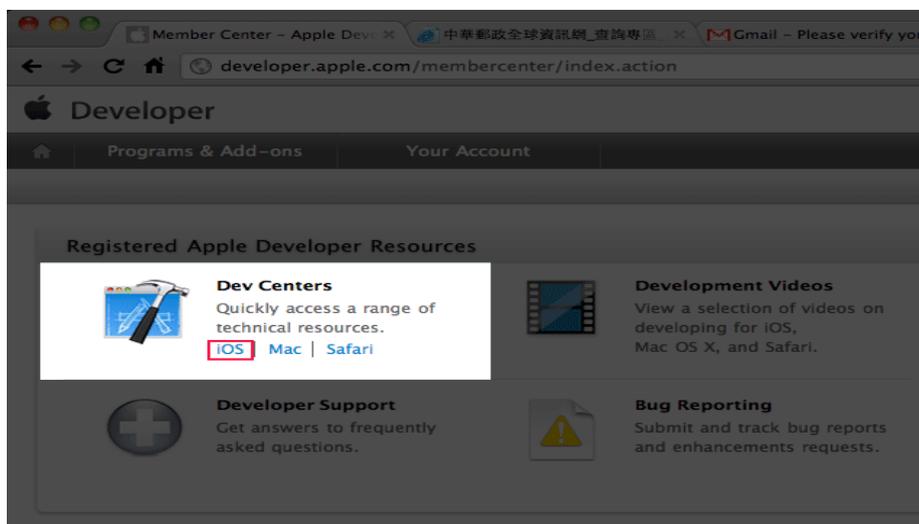


圖 39 Xcode 註冊畫面--9

## Step10

接著就會跳到Developer畫面，由於是要開發iPhone的應用程式，請點選iOS選項。



圖表 40 Xcode 安裝畫面--1

## Step11

接著再點選Download連結。

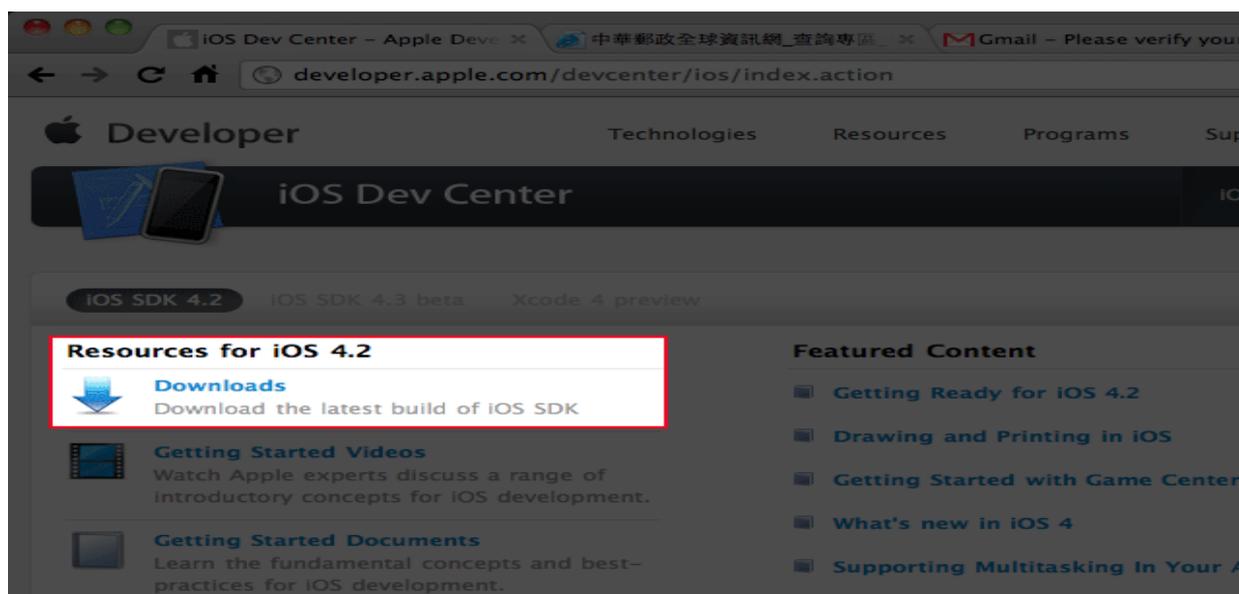


圖 41 Xcode 安裝畫面--2

## Step12

接到下方，點選*Xcode 3.2.5 and iOS SDK 4.2*的連結。

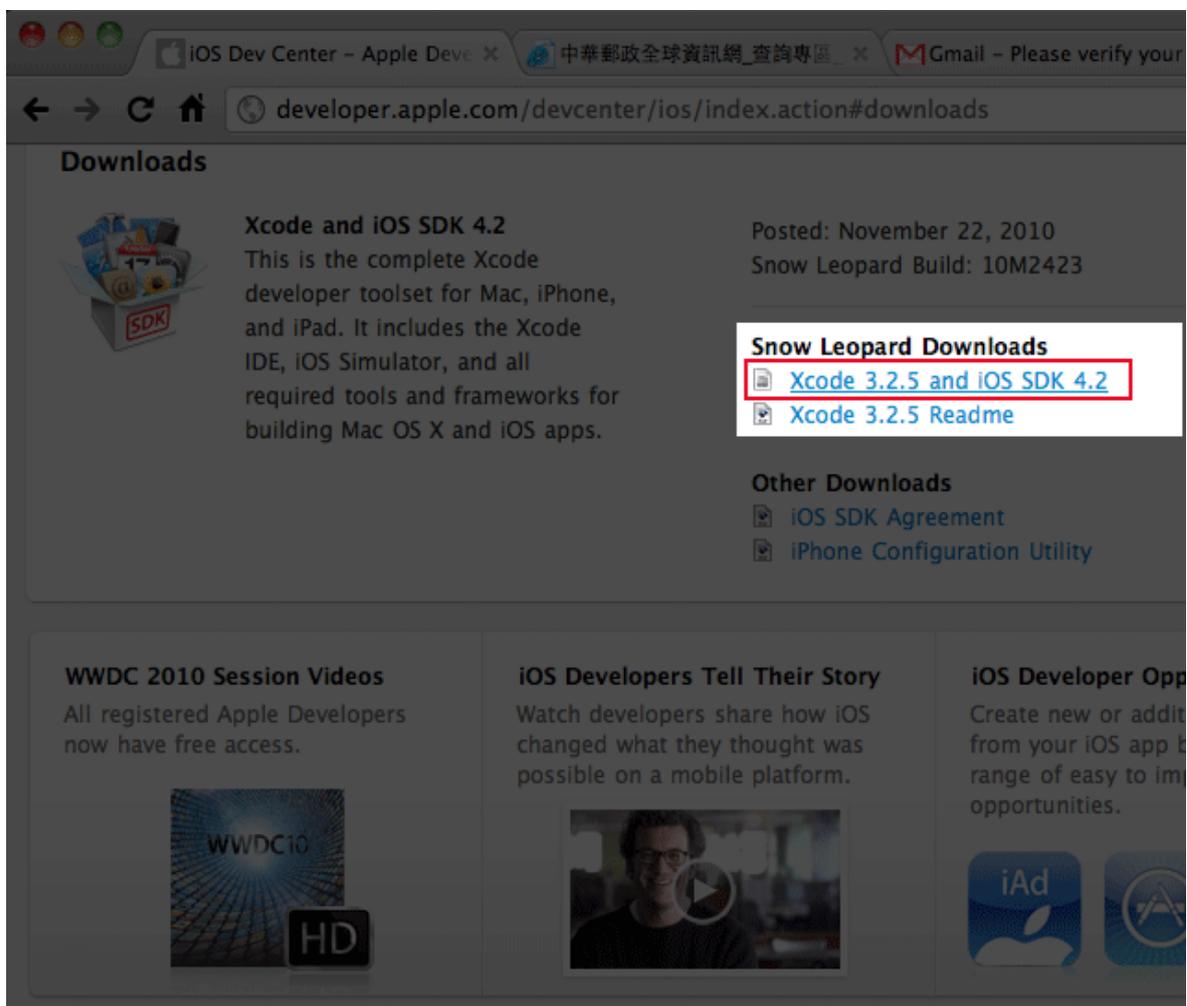


圖 42 Xcode 安裝畫面--3

## Step13

下載完畢後，雙響 *Xcode3.2.5 adn iOS SDK4.2.dmg* 檔，就會開始進行解壓並安裝。



圖 43 Xcode 安裝畫面--4

## Step14

解壓完成後，再雙響 *Xcode3.2.5 adn iOS SDK4.2.mpkg* 檔，並按繼續鈕。



圖 44 Xcode 安裝畫面--5

## Step15

接著就會出現安裝畫面囉!再按右下角的繼續鈕。



圖 45 Xcode 安裝畫面--6

## Step16

就一路一直下一步，就可完成安裝囉!但得注意到，一定要有9G的硬碟空間大小，否則會安裝失敗。

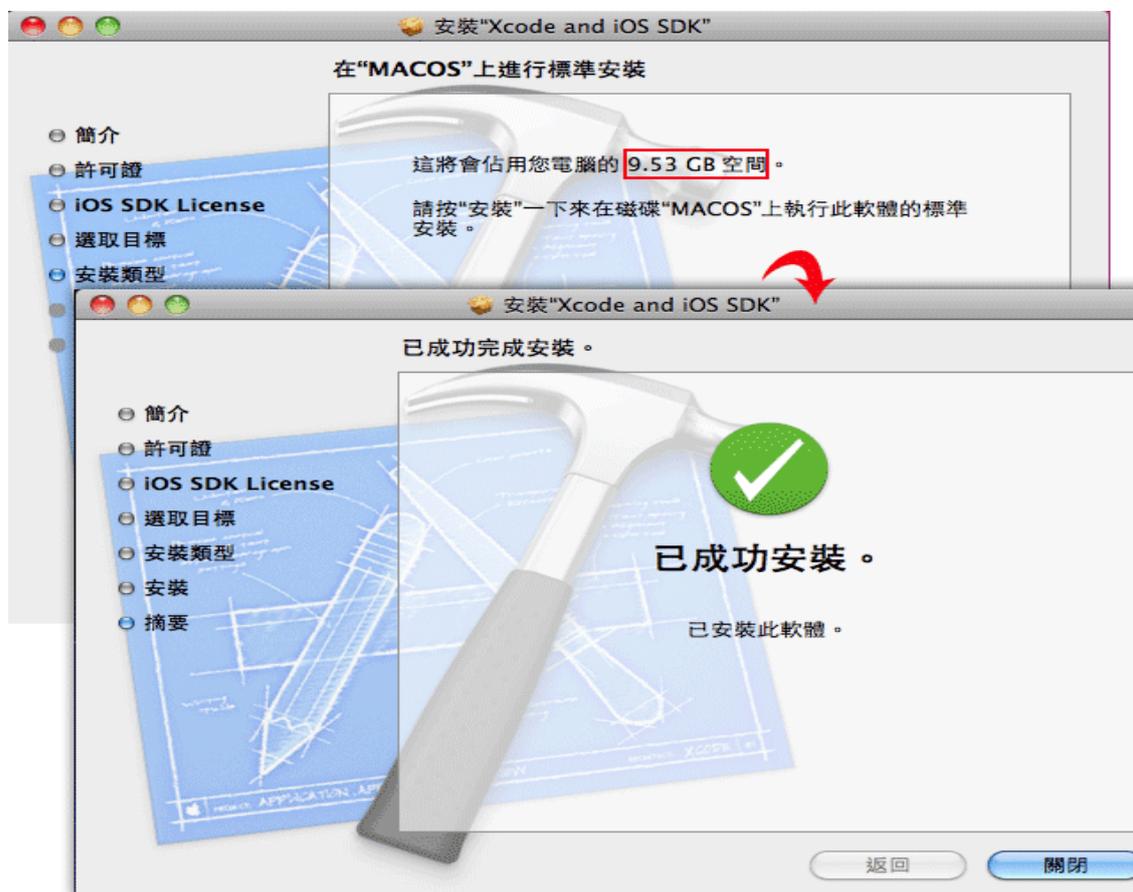


圖 46 Xcode 安裝畫面--7

## Step17

安裝後它並不會存放在應用程式，而是放在 磁碟 / Developer / Applications / Xcode.app。

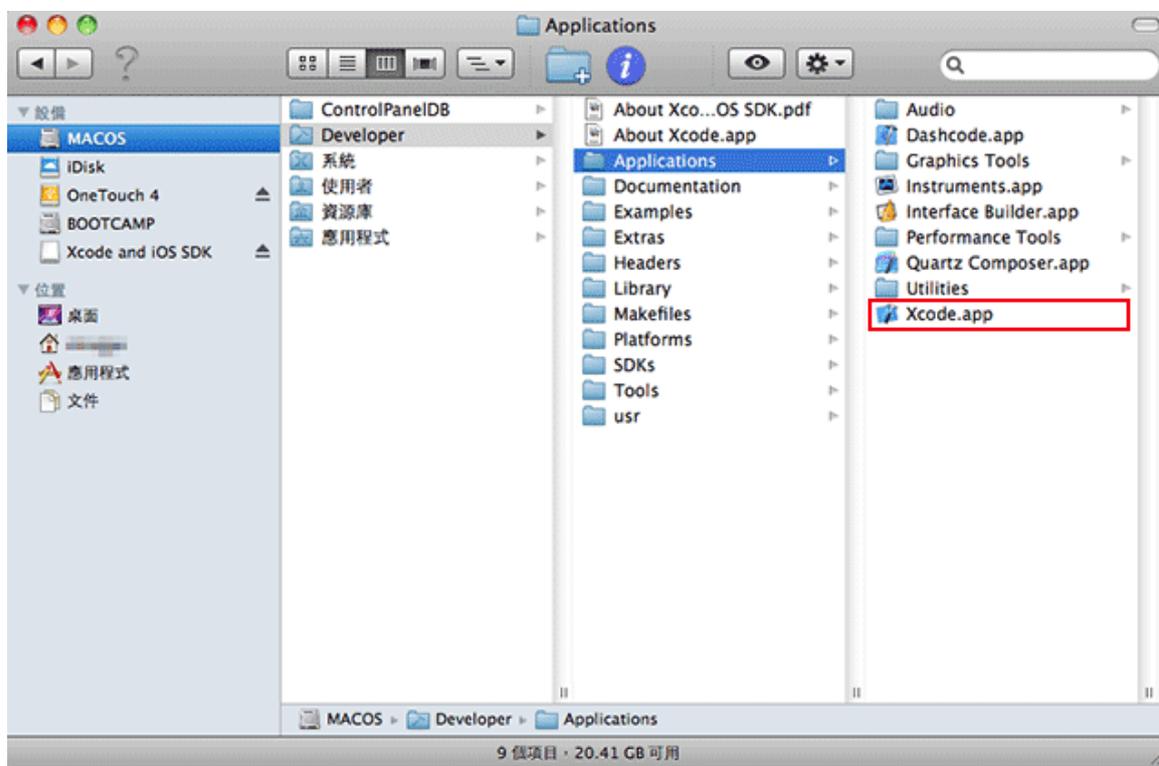


圖 47 Xcode 安裝畫面--8

## Step18

再雙擊Xcode. app後，將子就會開啟Xcode的開發畫面啦!接下來就交給有緣人自行研究囉!

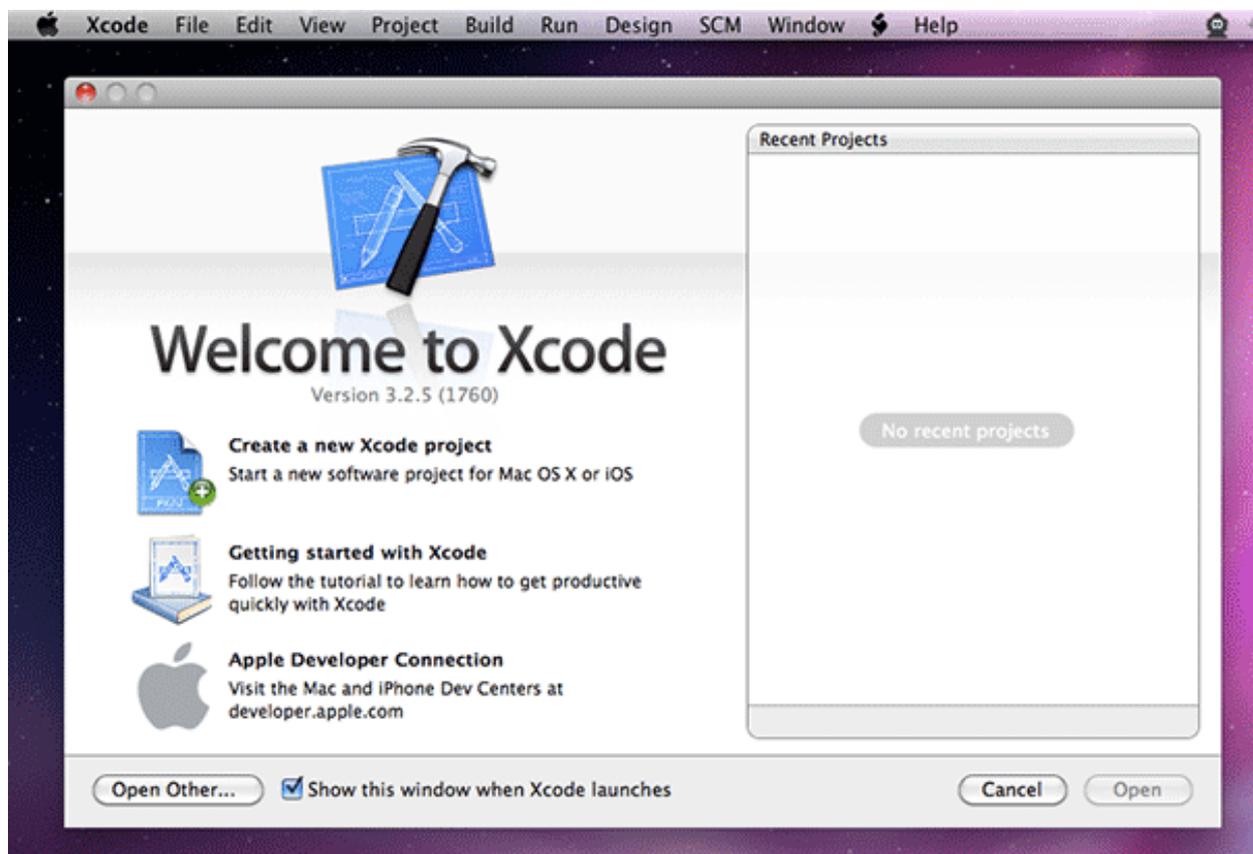


圖 48 Xcode 安裝畫面--9

這樣開發工具也有啦!對於想開發 iPhone/iPad 應用程式的朋友們，可以開始動工囉!

## 系統操作說明

首先我們要先開啓 Xcode

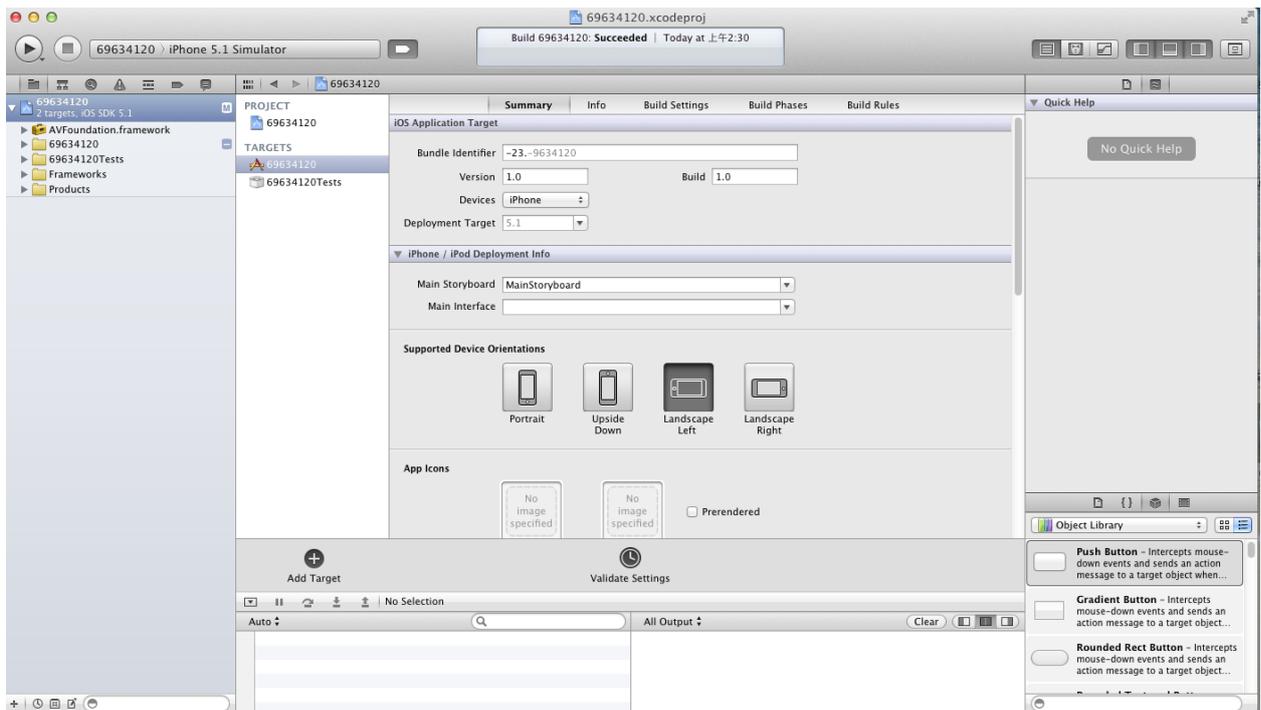


圖 49 系統操作說明--1

再來我們需要開啓 I O S 模擬器



圖 50 系統操作說明--2

開啓後會先跑一陣子之後就看到虛擬 iPhone



圖 51 系統操作說明--3

再來就要點選我們所作的程式囉



圖 52 系統操作說明--4

點進去之後就會看到整個英文字母學習介面



圖 52-1 首頁

點首頁進入後的畫面

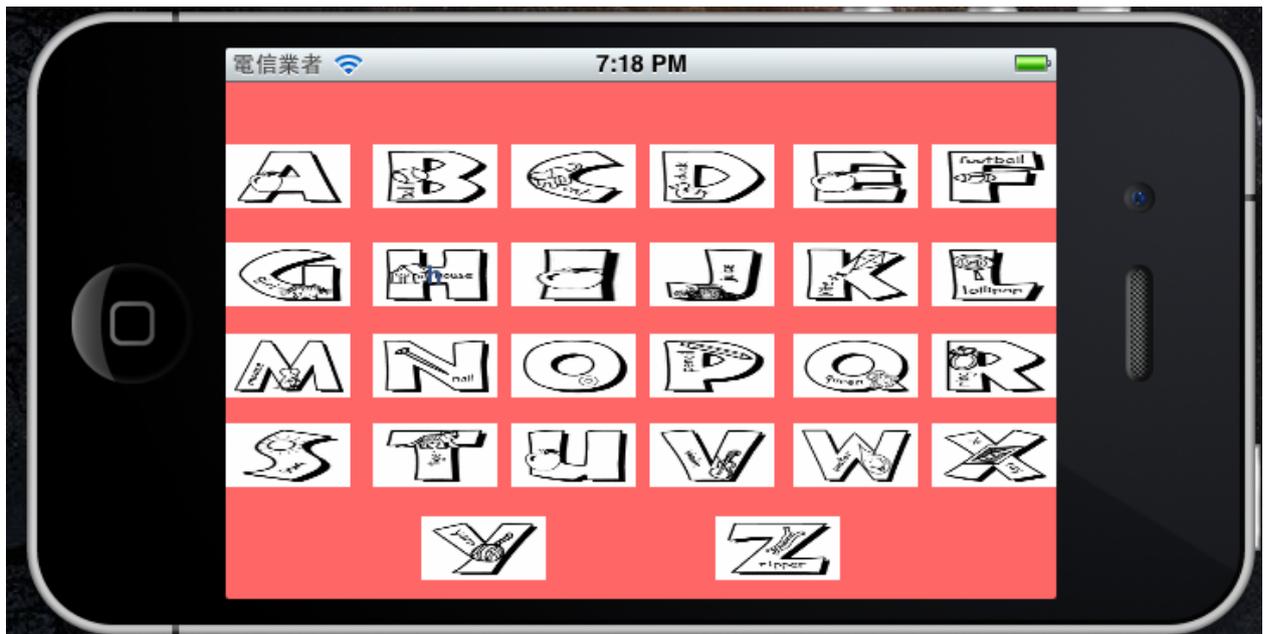


圖 53 系統操作說明--5

點擊字母就會跳出相關的



圖 54 系統操作說明--6



圖 55 系統操作說明--7



圖 56 系統操作說明--8

程式也設計好了，以後就可以上架了



圖表 57 系統操作說明--9

程式設計好後如果要開始上架去賣的話，首先你需要先跟蘋果申請繳年費。

可是那成本太粗了，對我這種剛入門的暫時不需要，等哪天比較成熟時再開始賺錢的路途吧。

## 第五章 結論

APP Store 這是塊非常龐大的市場，也因為這幾年來資訊的快速發展而開始有了更快速的成長，傳統手機經營者已經紛紛投入智慧型手機這種經營模式，以最快接近顧客的方式給予服務，而以網際網路達到全台各地甚至全世界來做買賣。

本專題的主要目的是讓各個下載我們軟體的人，可以在我們的軟體上找到親子同樂又可以教育下一代，透過手機或網際網路與智慧型手機的強大結合力，只要透過手機和網路，可以說全世界的人都有機會使用到我們的軟體。這樣的網路傳播力，對於產品的曝光非常有利，連帶的如果插入些廣告，又或者增加個商城或者加值軟體更可以賺進更大的利益率。也因為手機是幾乎人手一只，且又是因為網路的發達而減少了一些成本的支出（如店面租金、廣告行銷成本等），對價格的調整因此較具彈性。對消費者來說，透過網際網路連線，全天候二十四小時、不管在哪都能買到我們的產品，相對而言提高了對顧客的服務。由此可知透過蘋果的商店雖然要給他一小筆錢，但是卻省了時間與行銷！科技的不斷進步，它也徹底改變了人們傳統的手機使用觀念了。

## 參考文獻

- [1] App 程式設計入門 (松崗出版公司)
- [2] i OS sdk 開發範例大全(悅知出版公司)
- [3] Mac OS X Lion 10.7 我的第一本蘋果書 (碁峯出版公司)
- [4] 羅澂憑 - 跟 Adobe 徹底研究 Photoshop CS4 (上奇出版公司)
- [5] 博客來書店 - <http://www.books.com.tw/books/index.php>
- [6] 台灣蘋果 - <http://www.apple.com/tw/hotnews/>
- [7] yahoo 奇摩 - <http://tw.yahoo.com>
- [8] google - <http://www.google.com>