

# 高英高級工商職業學校

Kao Ying Industrial Commercial Vocational High School

## 教師專題研究（製作）報告



## 個人英文單字學習系統

老師姓名：\_\_\_\_\_簡琨祥\_\_\_\_\_老師

科 別：\_\_\_\_\_資訊科\_\_\_\_\_科

中 華 民 國 1 0 4 年 2 月

## 摘要

國際化語言學習為甚麼這麼迷人?當中的「商機」可說是非常驚人的，而現今是個國際化的時代，未來的發展也迅速的走向國際化，如何運用科技與生活語言學習之間的關係去將學習行為來做結合，並創造出龐大的商機，是我們所需要研究的。

從事教育專業領域的我，利用本身專業去研究語言學習的系統，讓每個人能夠體驗不一樣的學習感受，我以多元化的學習方式、便利性的操作設定，以及創新功能為研究目標。

研究導向：

1. 在學習英文的過程中，必須使用許多的學習方法及時間來了解英文單字的意思，出外時其實有許多的時間可以利用來學習英文，可是都因為身邊沒有任何可以學習的教材所以無法學習，因為隨時攜帶書本是一個負擔，如何讓不用攜帶書本又能在出外之間學習來獲得相應的學習效果。
2. 練習單字之後或多或少會遇到一再遺忘的問題，針對學習不中斷及反覆練習，我們可以在任何的時間點來進行自我的英文學習。
3. 藉由平常生活及旅遊所接觸到的英文單字，將之做到資料庫，並增加自助旅遊時可以用到。

總結以上三點研究方向，除了讓英文單字學習更有導向性外，並添加自我單字測驗的挑戰樂趣，後續如能夠繼續完全開發，將會在商業英文或旅遊英文這領域帶來不可預知的一股新學習潮流及未知的商機發展。

# 目 錄

目錄.....	iii
表目錄.....	iv
圖目錄.....	iv
壹、前言.....	1
一、製作動機.....	1
二、研究目的.....	1
三、製作架構.....	2
四、製作預期成效.....	3
貳、理論探討.....	4
一、研究分析.....	4
參、專題製作過程或方法.....	17
一、製作設備及器材.....	17
二、製作方法與步驟.....	18
三、專題製作.....	22
四、製作成果.....	23
肆、製作結論與建議.....	26
一、結論.....	26
二、建議.....	26
伍、參考文獻.....	27

## 表目錄

表 1-1-3-1 平板電腦各作業系統比例.....	7
表 1-1-3-2 用戶使用系統版本比例.....	7
表 3-1-1 專題製作使用儀器（軟體）設備一覽表.....	17

## 圖目錄

圖 1-3-1 語言轉換過程圖.....	1
圖 1-3-2 專題製作架構圖.....	2
圖 2-4-1 APP 軟體釋義圖(一).....	7
圖 2-4-2 APP 軟體釋義圖(二).....	7
圖 2-5-1 使用者介面.....	9
圖 2-5-2 版面佈局.....	10
圖 2-5-3 影音媒體.....	11
圖 2-5-4 繪圖與動畫.....	11
圖 2-5-5 預覽視窗.....	12
圖 2-5-6 組成元件.....	12
圖 2-5-7 屬性面板.....	13
圖 2-5-8 媒體庫.....	13
圖 2-5-9-1 Blocks.....	14
圖 2-5-9-2 Blocks 種類(一).....	14
圖 2-5-9-3 Blocks 種類(二).....	15
圖 2-5-10 程式出錯警告區.....	15
圖 2-5-11 垃圾桶.....	16
圖 2-5-12-1 相關書籍(一).....	16
圖 2-5-12-2 相關書籍(二).....	16
圖 2-5-12-3 相關書籍(三).....	16
圖 2-5-12-4 相關書籍(四).....	16
圖 3-2-1 製作方法與步驟.....	18
圖 3-2-2 County.....	19
圖 3-2-3 Language.....	19
圖 3-2-4 Result.....	19
圖 3-2-5 Pitch.....	20
圖 3-2-6 Speak(Textmessage).....	20

圖 3-2-7 AfterSpeaking(Text result) .....	20
圖 3-2-8 BeforeSpeaking .....	20
圖 4-1-1 SpeakInEnglish_4 啟動.....	23
圖 4-1-2 SIE_4 軟體初始畫面.....	23
圖 4-1-3 SIE_4 單字庫.....	23
圖 4-1-4 SIE_4 單字練習_選單字.....	23
圖 4-1-5 SIE_4 單字練習_字義.....	23
圖 4-1-6 SIE_4 單字練習_發音.....	23
圖 4-2-1 SIE_4 新增單字_輸入 .....	24
圖 4-2-2 SIE_4 新增單字_儲存.....	24
圖 4-3-2 SIE_4 刪除單字_確定.....	24
圖 4-3-3 SIE_4 刪除單字_刪除鈕 .....	24
圖 4-3-4 SIE_4 刪除單字_檢查字庫.....	24
圖 4-4-1 SIE_4 單字測驗_隨機選擇 .....	25
圖 4-4-2 SIE_4 單字測驗_檢測單字.....	25

# 壹、前言

## 一、製作動機

現在的學生，對於未來需要很多的準備，外國語言可說是最重要的能力之一，然而多種國家的語言裡面最常見的就是英文，英文的國際化帶來許多商機，有一次我看著英文講義並對它產生了一些疑問？，因而想起英文單字 20000 的教科書擁有很多的單字與例句，吸引了許多的學生的注意，但過了不久，缺點慢慢出現，在於它單純的呈現單字但是數量眾多而無法及時的找到想要的單字的需求，並且攜帶不便，它終究不能隨身攜帶，無法讓學生達到即時學習的功效，而這也是它最大的「遺憾」，因為書本特點就在於它的訊息眾多但是卻攜帶不便，現今雖然有許多的翻譯機的產品出現，但市面上販售的翻譯機或英文辭典價格昂貴，如果我們只需用到一些需要的英文單字，像是英文檢定只需要一定的英文單字，而且在攜帶上也有額外的負擔及失竊的風險。而我經過那次的疑問後想出了許多解決方法，藉由這次研究希望可以改善書本及市售翻譯機的缺點，並且結合上述兩者的優點來製作出適合的成品。

## 二、研究目的

現今的讀書方式依然回歸到翻書或是使用翻譯機的時代，而近年來也有許多嘗試精簡化書本的訊息來達到便利的效果，但是過於精簡的內容達不到預期的效用。

而若是使用翻譯機，必須隨身的攜帶才可以，而每次的攜帶都有可能造成遺失或損壞，而且翻譯機主要的問題在於文句中所包含句法、語義成分較低，因而在處理句法差別較大的語言例如中文-英文時將

遇到問題。這時翻譯結果雖然「詞詞都對」卻無法被人閱讀理解。可以說目前翻譯機無法有效改善的問題。這時如果有一個可以讓使用者可以自行編義的翻譯機的話就可以人性化的去解釋詞句了，並且如果它能夠隨身攜帶，能夠達到隨時使用的要求，就能夠省下很多的時間及金錢，並能增學習的成效。

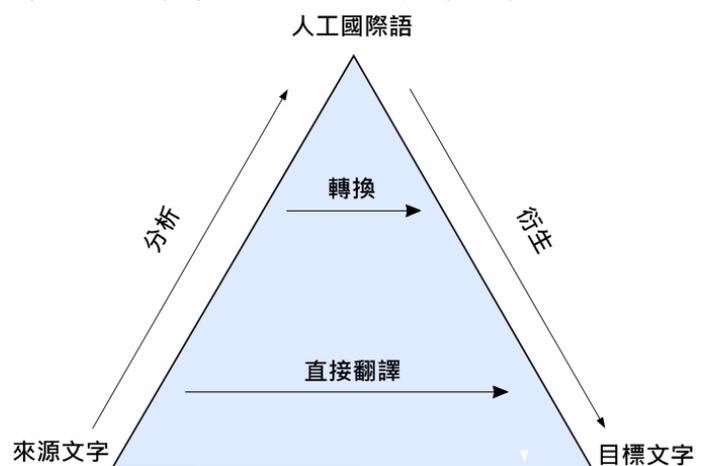


圖 1-3-1 語言轉換過程圖

### 三、製作架構

#### (一)、專題製作架構圖

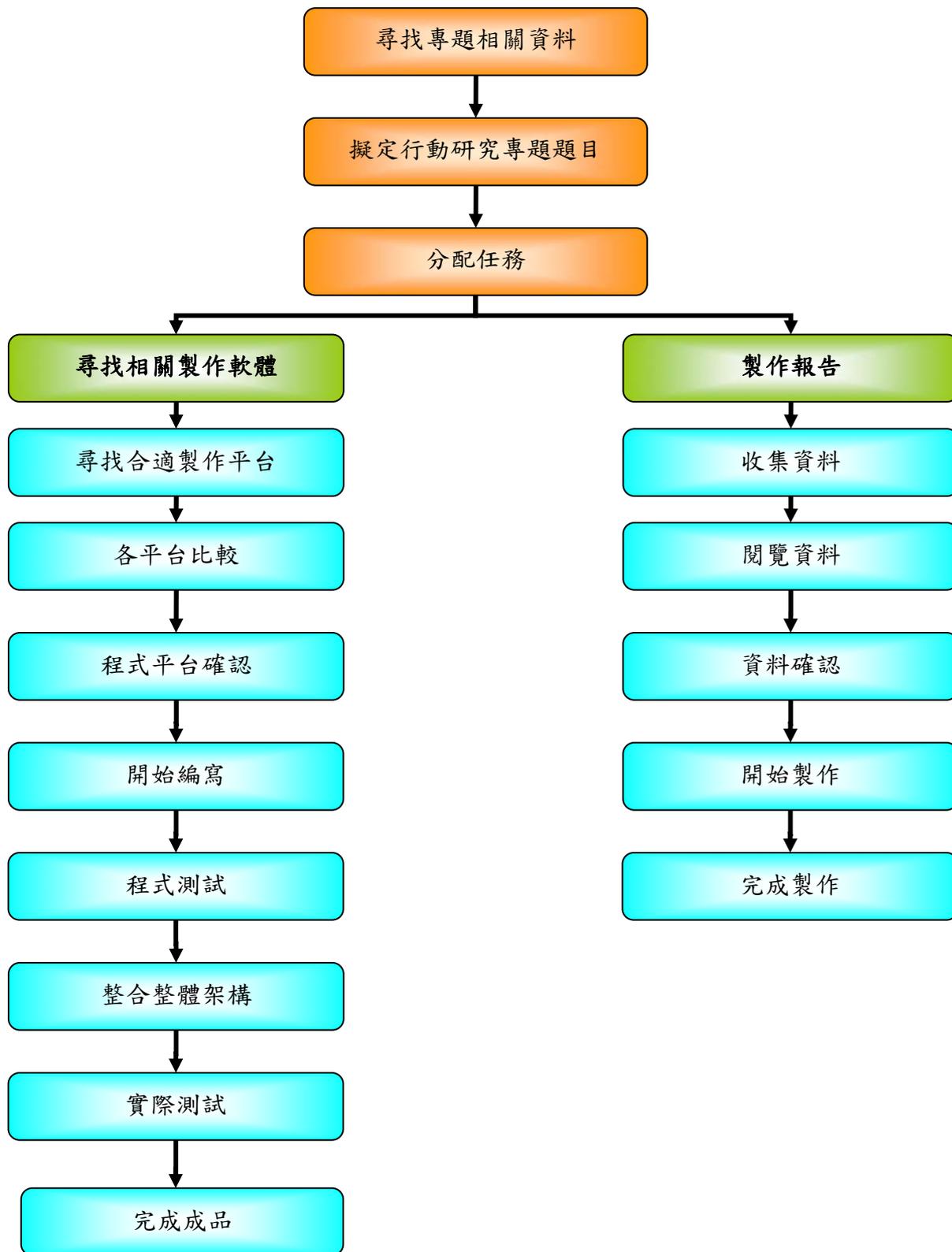


圖 1-3-2 專題製作架構圖

#### 四、製作預期成效

本專題預計實做完成符合現代商品化需求的「翻譯系統」。此專題配合專業課程的理論內容及實習技能，同時利用此行動研究專題培養自我的科技研究精神。且翻譯程式之主要改善書本及市售翻譯機的缺點及以創意來創造新產品，進而商品化。本次翻譯系統製作將達到以下之目標：

- (一)、減少龐大資料庫的搜尋時間。
- (二)、增加攜帶的便利性。
- (三)、滿足使用者的『彈性化』，『客製化』的需求。
- (四)、以自己的方式來讓使用者學習。
- (五)、支援英文語音發音，可練習聽力及正確發音。
- (六)、有效的達到市場的商品競爭力。

## 貳、理論探討

### 一、研究分析：

#### (一)、App Inventor2 專案研究簡介

現在的科技日新月異，手機更是人手一支，而每隻手機當中最中要的便是 App (Application)，而依個人想要去撰寫這種程式，或者從事程式是開發的話，那麼就必須要知道 App Inventor2、Android，以及開程式所需要的軟體，還有開發所需要的環境，因此在本章節中會詳細的介紹。

#### (二)、什麼是 APP？

App 是「Application」的縮寫，就是手機「應用程式」「應用軟體」的意思。

近年來 App 這個字眼充斥在我們生活中，原因是智慧型手機越來越普及。智慧型手機的 App 就如同電腦中的各種軟體，只是作業系統不同、內容相對簡單罷了。

「智慧型手機」(Smart Phone)這個說法主要是針對「功能手機」(Feature phone)，作為一個區隔而來，並非這個手機有多麼「智慧(Smart)」。傳統的功能手機的操作模式與內容是被手機廠限制固定的，不能由使用者隨意安裝移除軟體。而智慧型手機有提供一個平台空間，讓使用者可以隨意安裝和移除應用軟體 (App)，就像電腦那樣。因此智慧型手機可定義為：擁有開放系統環境，允許第三方自行研發 App，並提供使用者自由運用的手機。

智慧型手機作業系統(OS)的四巨頭為 Apple 的 IOS、Google 的 Android、Microsoft 的 Windows Mobile (最近推出 Win8 RT)、RIM 的 Black Berry，其他還有 Nokia (Symbian)、Palm 等。不同作業系統的應用程式無法互通，因為內部程式指令與運作方式各有不同。

### (三)、Android 系統簡介：

網路巨人 Google 於 2007 年 11 月 5 日推出 Android，Android 是一套建構在 Linux 核心(Linux Kernel)之上的智慧型手機作業系統，Android 作業系統的核心屬於 Linux 核心的一個分支，具有典型的 Linux 排程和功能，除此之外，Google 為了能讓 Linux 在行動裝置上良好的運行，對其進行了修改和擴充。Android 去除了 Linux 中的本地 X Window System，也不支援標準的 GNU 庫，這使得 Linux 平台上的應用程式移植到 Android 平台上變得困難。2008 年，Patrick Brady 於 Google I/O 演講「Anatomy & Physiology of an Android」，並提出的 Android HAL 架構圖。HAL 以\*.so 檔的形式存在，可以把 Android framework 與 Linux kernel 隔開，這種中介層的方式使得 Android 能在行動裝置上獲得更高的執行效率。這種獨特的系統結構被 Linux 核心開發者 Greg Kroah-Hartman 和其他核心維護者稱讚。Google 還在 Android 的核心中加入了自己開發製作的一個名為「wakelocks」的行動裝置電源管理功能，該功能用於管理行動裝置的電池效能，但是該功能並沒有被加入到 Linux 核心的主線開放和維護中，因為 Linux 核心維護者認為 Google 沒有向他們展示這個功能的意圖和代碼。

2010 年 2 月 3 日，由於 Google 在 Android 核心開發方面和 Linux 社群方面開發的不同步，Linux 核心開發者 Greg Kroah-Hartman 將 Android 的驅動程式從 Linux 核心「狀態樹」(「staging tree」)上除去。2010 年 4 月，Google 宣布將派遣 2 名開發人員加入 Linux 核心社群，以便重返 Linux 核心。2010 年 9 月，Linux 核心開發者 Rafael J. Wysocki 添加了一個修復程式，使得 Android 的「wakelocks」可以輕鬆地與主線 Linux 核心合併。2011 年，Linus Torvalds 說：「Android 的核心和 Linux 的核心將最終回歸到一起，但可能不會是 4-5 年。」在 Linux 3.3 中大部分代碼的整合完成。]

在早期的 Android 應用程式開發中，通常通過在 Android SDK (Android 軟體開發包) 中使用 Java 作為編程語言來開發應用程式。開發者亦可以通過在 Android NDK (Android Native 開發包) 中使用 C 語言或者 C++ 語言來作為編程語言開發應用程式。同時 Google 還推出了適合初學者編程使用的 Simple 語言，該語言類似微軟公司的 Visual Basic 語言。此外，Google 還推出了 Google App Inventor 開發工具，該開發工具可以快速地構建應用程式，方便新手開發者。

Android 作業系統使用了沙箱 (sandbox) 機制，所有的應用程式都會先被簡單地解壓縮到沙箱中進行檢查，並且將應用程式所需的權限提交給系統，並且將其所需權限以清單的形式展現出來，供用戶檢視。例如一個第三方瀏覽器需要「連接網路」的權限，或者一些軟體需要撥打電話，發送簡訊等權限。用戶可以根據權限來考慮自己是否需要安裝，用戶只有在同意了應用程式權限之後，才能進行安裝[。

由於，Android 在軟體版本授權上是採用 Apache Software License 2.0 的開放原始碼方案，因此在這個版權協議之下，智慧型手機製造商可免費地安裝 Android 作業系統至其生產製造的硬體之中，有效地降低了軟體的採購成本。

目前 Android 版本已經來到 5.0. x Lollipop，另有其他的舊版本，請參考相關網頁 <http://developer.android.com/>。

對於智慧型手機製造商來說，透過免費取得作業系統而降低軟體採購成本是一項很大的誘因，所以吸引眾多廠商投入生產、銷售或者研發 Android 作業系統的相關軟硬體產品與服務。Android 以新秀之姿在短短五年的時間就有此成績，不只對於旗下合作的智慧型手機製造商具鼓舞作用，也會促使 Android 應用程式的開發者，投入更多時間與精力去開發兼具功能性與創新性的應用程式。再更進一步地說，由於 Android 在應用程式開發上，採取免費、開放的策略。開發者不僅可以免費地下載安裝 Android SDK (Android 的軟體開發工具包) 進行應用程式的開發。更重要地是，人們可以使用多數程式設計師所熟悉的 Java 程式語言進行應用程式的編寫。因為這兩個特點，促使為數眾多的 Java 程式設計師蜂擁至 Android 應用程式的開發行列。

## 1. 平板電腦各作業系統比例

表 1-1-3-1 平板電腦各作業系統比例

作業系統	2013 銷售量	2013 市佔率	2012 銷售量	2012 市佔率
Android	120,961,445	61.9%	53,341,250	45.8%
iOS	70,400,159	36.0%	61,465,632	52.8%
Windows	4,031,802	2.1%	1,162,435	1.0%
Other	41,598	0.1%	379,000	0.3%

## 2. 用戶使用系統版本比例

Android 作業系統版本的用戶比例（2014 年 8 月 14 日）截止至 2014 年 12 月 1 日，不同的 Android 作業系統版本的用戶比例。

表 1-1-3-2 用戶使用系統版本比例

版本名稱	API level	用戶分佈
5.0.x Lollipop 棒棒糖	21	50.9%
4.4 KitKat 奇巧巧克力	19	33.9%
4.3 Jelly Bean 雷根糖	18	7.0%
4.2.x Jelly Bean 雷根糖	17	20.4%
4.1.x Jelly Bean 雷根糖	16	21.3%
4.0.3-4.0.4 Ice Cream Sandwich 冰淇淋三明治	15	7.8%
3.2 Honey Comb 蜂巢	13	N/A
2.3.3-2.3.7 Ginger Bread 薑餅	10	9.1%
2.2 Froyo 霜凍優格	8	0.5%

#### (四)、App Inventor2 簡介：

Google 在 2010 年底推出 App Inventor2 給所有使用者使用，App Inventor2 由一群 Google 工程師與勇於挑戰的 Google 使用者共同參與開發。Google App Inventor2 是一個完全線上開發的 Android 程式環境，Android SDK 環境建置在雲端(網路上)，使用者只要在 PC 上安裝 JAVA 即可立即使用。App Inventor2 拋棄複雜的程式碼而使用積木式的堆疊法來完成 Android App 程式的設計。除此之外它也正式支援樂高 NXT 機器人，對於 Android 初學者或是機器人開發者來說是一大福音。因為對於想要用手機控制機器人的使用者而言，他們不大需要太華麗的介面，只要使用基本元件例如按鈕、文字輸入輸出即可。

由於是針對學生族群，Google 在這個部分特別引入在教育系統當中頗為成功的 Scratch 來作為拼圖式程式寫作的基礎，標榜可以讓使用者完全不需寫任何程式碼即可完成應用程式開發。更令人驚豔的是，支援 NXT 樂高機器人程式開發，使得 Android 與 NXT 結合應用，增添了許多想像空間。

由於 App Inventor2 突破性的發展，使得原本可能是處於高階程式設計師領域的手機應用程式開發，一下子成為非專業人員甚至是國小學童接觸程式設計的入門起步。

不過，Google Lab 已於 2011 年停止了繼續發展的計畫，並於 2012 年 1 月 1 日將所有資源移交給麻省理工學院行動學習中心，並以 MIT App Inventor2 名稱繼續提供服務。



圖 2-4-1 APP 軟體釋義圖(一)

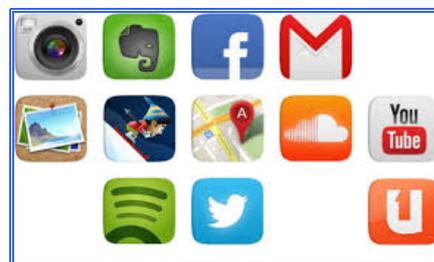


圖 2-4-2 APP 軟體釋義圖(二)

### (五)、App Inventor2 環境建置

Google App Inventor2 為全雲端的開發環境，所有的動作皆在瀏覽器上完成（建議使用 Google Chrome），開發設計資料全放在 MIT App Inventor2 伺服器上。

您需要三件事：

1. Gmail 帳號（如果沒有 Gmail 帳號，請到 google 網站新申請一個。）
2. Java, 測試電腦上是否可正確運行 Java(測試頁面)。如果尚未安裝 java 或版本不適合，請依照指示下載並安裝新的版本。
3. 安裝 Google App Inventor2 軟體。

接著便能開始從事研發的工作，只要將版面設計完成，並且加上運作所需的程式，就能使一個簡單的應用程式運作；不過這只是相單簡略的應用程式，若是想要完成更艱深的應用程式，可以參考網路上的範本。



## (二) 使用者介面(User Interface)

使用者介面是最經常被使用的一個子項目，當中包括按鈕、圖片、文字輸入格等基本元件。



圖 2-5-1 使用者介面

## (三) 版面佈局(Layout)

當我們設計手機遊戲時，常需要使用不同的排版，增加畫面的豐富性。App Inventor 2 本身的排版是垂直的(Vertical)，因此當我們想要變化物件的排版方式時，便需要使用版面佈局(Layout)的元件。

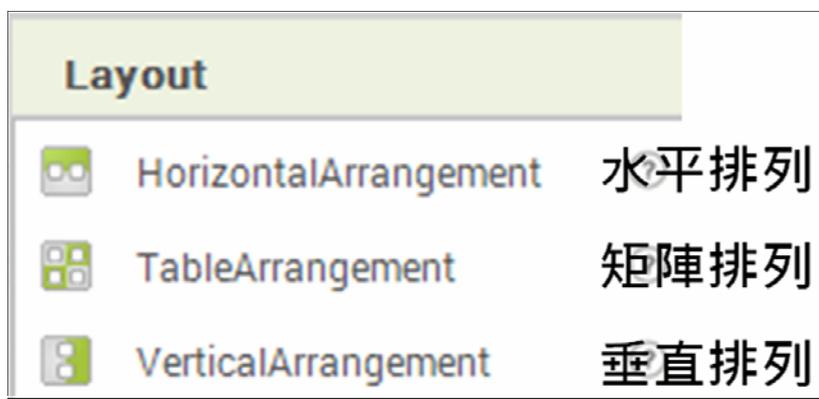


圖 2-5-2 版面佈局

#### (四) 影音媒體(Media)

加入影音媒體可以讓你的程式增加更多互動性與趣味。

Media		
	Camcorder	攝影
	Camera	拍照
	ImagePicker	挑選圖片
	Player	聲音播放器
	Sound	聲音
	SoundRecorder	錄音
	SpeechRecognizer	聲音辨認
	TextToSpeech	語音文字
	VideoPlayer	影片播放器

圖 2-5-3 影音媒體

#### 1. 繪畫與動畫(Drawing and Animation)

要讓你的程式裡有動畫，那就要使用這一個子項目，運用「畫布(Canvas)」這個元件當作框架開始設計你的動畫

Drawing and Animation		
	Ball	球
	Canvas	畫布
	ImageSprite	圖片動畫

圖 2-5-4 繪圖與動畫

## (2) 預覽視窗(Viewer)

預覽視窗就像我們的手機，在這裡設計你的手機畫面。

操作方式：在第一個區塊元件面板(Palette)選取元件後，將元件拖曳至預覽視窗(Viewer)中。

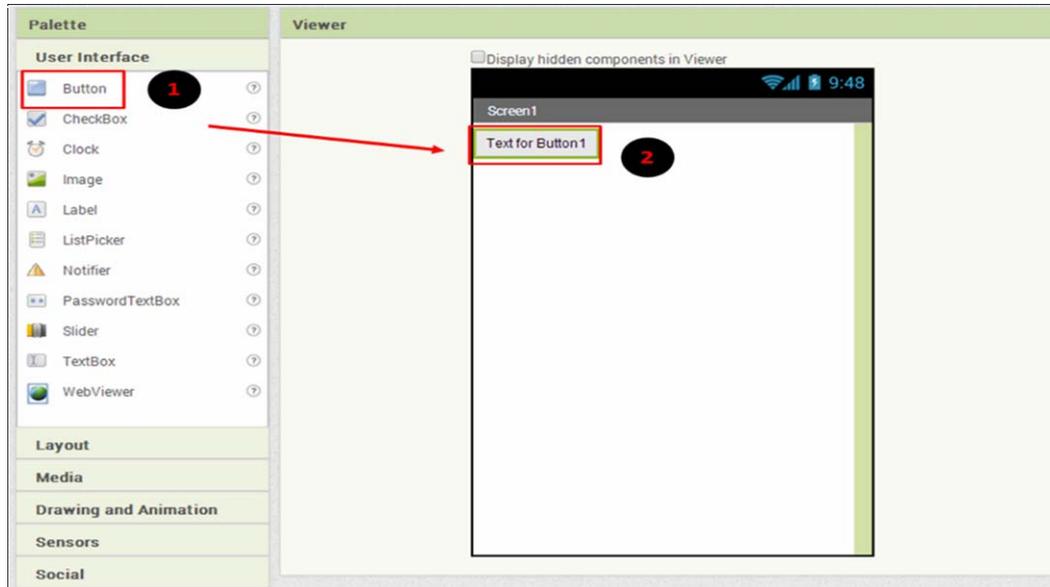


圖 2-5-5 預覽視窗

## (3) 組成元件(Components)

在這裡你可以管理你所使用的元件



**Rename**：重新命名元件，設計程式指令時較易辨別。

**Delete**：刪除元件。

圖 2-5-6 組成元件

#### (4) 屬性面板(Properties)

在屬性面板中(Properties)你可以調整你的元件屬性，如下圖以按鈕(Button)作範例說明。



圖 2-5-7 屬性面板

#### (5) 媒體庫(Media)

在這裡你可以一次上傳所有你需要的媒體素材，包含圖片、音樂。

(注意：檔名中可用底線，但不能有空格或其他特殊字元，且必須含有英文字)

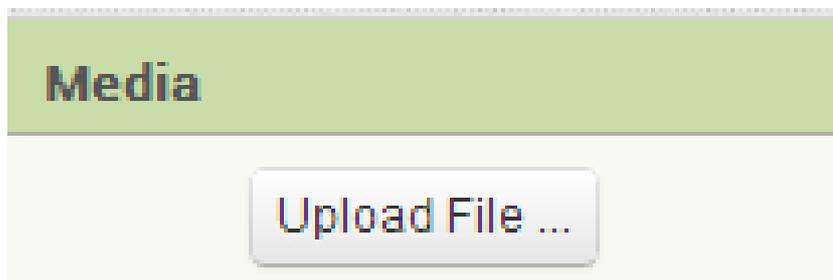


圖 2-5-8 媒體庫

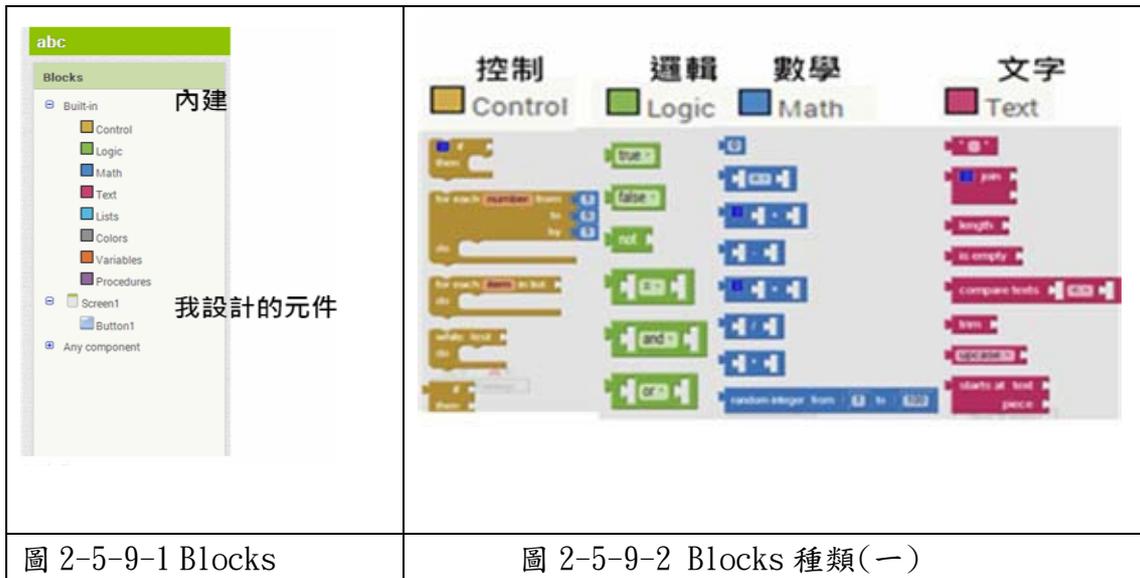
### (1) Blocks 物件

Blocks 分為兩大區塊，一個是內建、另一個是你自己設計的元件。

裡面有 App Inventor 2 幫你寫好的基本語法，只要找到你要的，將它拖曳出來就可以囉！

內建 Blocks 又分成了八種

- a. 控制(Control)：控制 if-then 的程序性問題、迴圈設計
- b. 邏輯(Logic)：包含 true、false 等，判斷性問題
- c. 數學：純數值、數學的四則運算、亂數設定
- d. 文字：字串、使字串相連



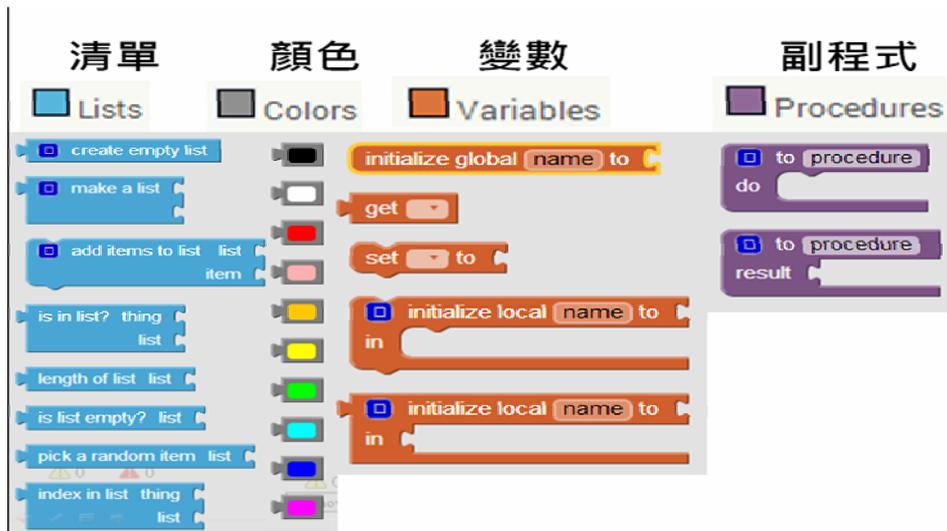


圖 2-5-9-3 Blocks 種類(二)

- e. 清單(List)：
  - 產生表格、提供暫時儲存空間
- f. 顏色(Color)：
  - 設定顏色
- g. 變數(Variables)：
  - 設定像 X、Y 的變數
- h. 副程式(Procedures)：
  - 需要重複執行的程式可以利用副程式縮短程式

**(3) 程式出錯警告區**

App Inventor 2 會自動幫你檢查，如果你的程式出錯了，它會像圖中一樣有警告標誌，告訴你哪裡有錯要修改。

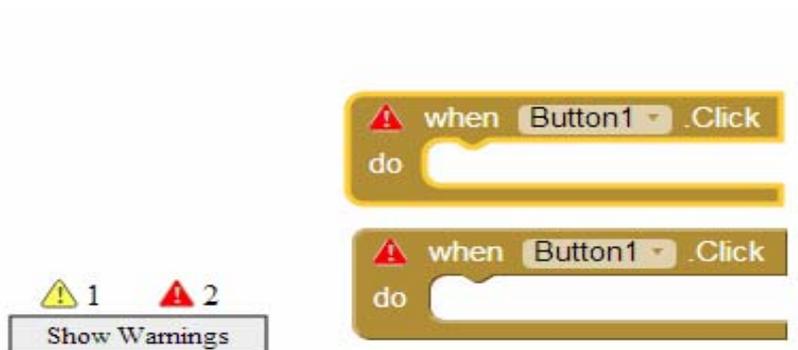


圖 2-5-10 程式出錯警告區

#### (4) 垃圾桶

將不需要或是錯誤的程式拼圖刪除

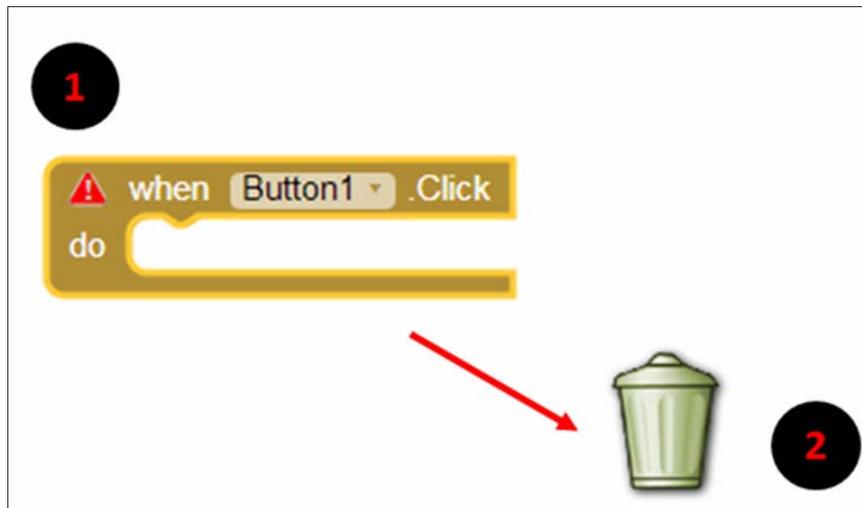


圖 2-5-11 垃圾桶

## 參、專題製作過程或方法

### 一、製作設備及器材

表 3-1-1 專題製作使用儀器（軟體）設備一覽表

儀器（軟體） 設備名稱	應用說明
個人電腦	專題報告撰寫、電路圖繪製及專題成品測試
Google Chrome	操作 APP Inventor 2
APP Inventor 2	程式設計
數位相機	紀錄整個專題製作流程
WIN XP	作業系統
噴墨印表機	列印專題相關資料
Microsoft Office Word	製作專題報告
ASUS MEMOPAD 平板電腦	操作 Android 系統控制 APP 程式

## 二、製作方法與步驟

製作「個人英文單字學習系統」專題，使用個人電腦利用 MIT 開發的 APP Inventor 2 來撰寫程式 APP 應用程式。所以除了程式設計需要有一定的能力之外，還需要一些基本的英文能力。

所以，為達到專題預期的目標，我們專題製作採用實作研究的方法，先透過題目的問題陳述、討論及訂出軟硬體規範與預期目標，在翻譯程式系統、面板設計及錯誤修正（實驗觀察）等步驟繼而完成本專題預定目標。經過上述的研究分析後，本專題研究之製作方法與步驟如下圖。

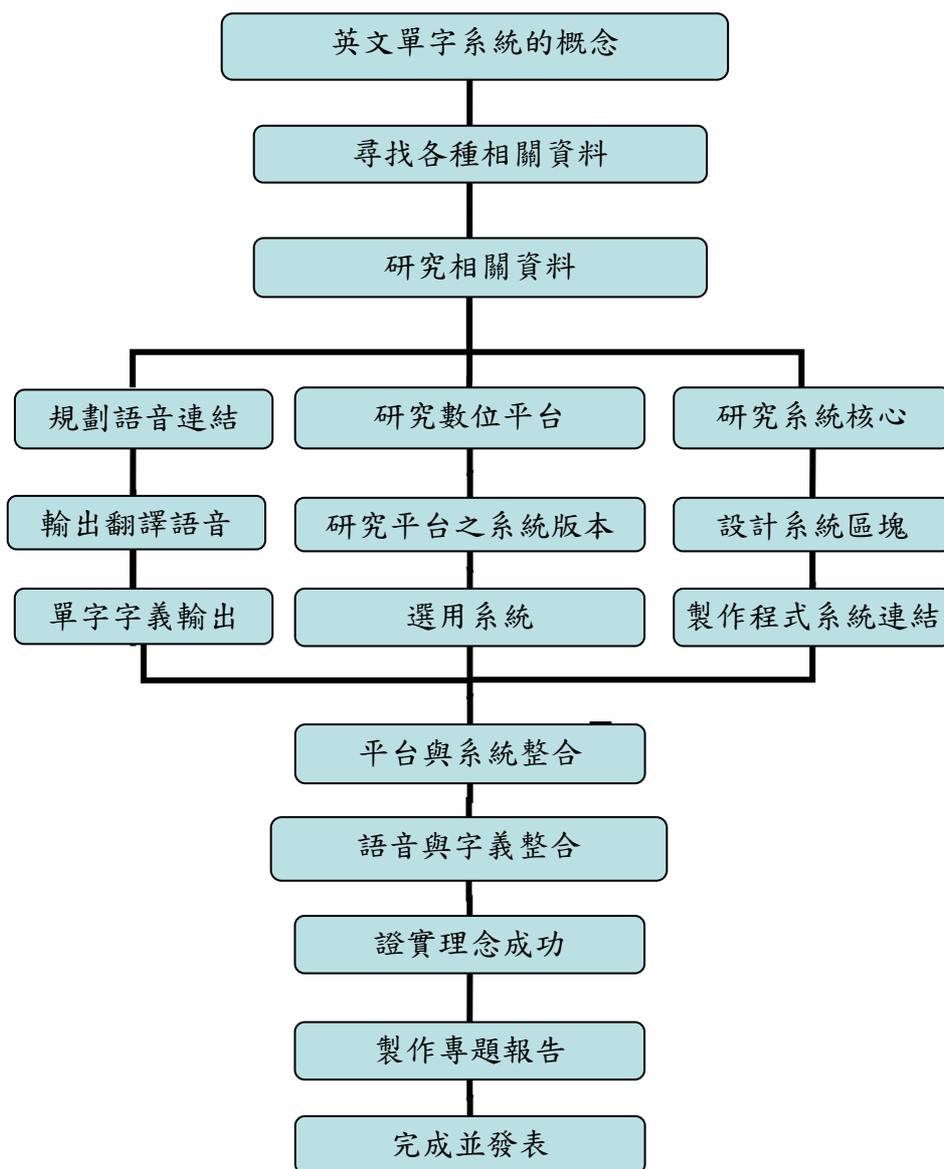


圖 3-2-1 製作方法與步驟

### (一)、App 文字轉語音程式

使您的裝置能念出文字資料。要使這項元件得以運作，您的裝置必須要裝有

Eyes-Free

Project 建置的 TTS Extended Service app，在

<http://code.google.com/p/eyes-free/downloads/list> 可以下載得到。

文字－語音轉換元件有語音文字轉換的相關屬性設定，通常是以三個字母的代碼來表示語言以及使用區域。

舉例而言，您可以區分英式與美式英語，英式英語的語言代碼為 eng，區域代碼則是 GBR；而若是美式英語，語言代碼依舊是 eng，而區域代碼則是 USA。後面會列出完整的列表。

#### ► Country

TextToSpeech.Country：語音輸出的國家代碼。



圖 3-2-2 Country

#### ► Language

TextToSpeech.Language：語音輸出的語言代碼。



圖 3-2-3 Language

#### ► Result

TextToSpeech.Result：詳細資訊。

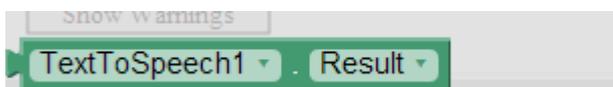


圖 3-2-4 Result

### ► Pitch

TextToSpeech.Pitch : 音調。

設定 TTS 的音調高低，數字介於 0~2，值越低音調越低，反之則越高



圖 3-2-5 Pitch

### ► Speak(Text message)

TextToSpeech.Speak : 發送指定文字資料。

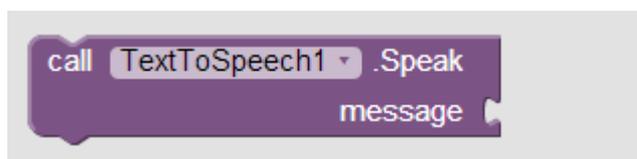


圖 3-2-6 Speak(Textmessage)

### ► AfterSpeaking(Text result)

TextToSpeech.AfterSpeaking : 文字資料發送後的信號，該參數是依文字資料的結果產生的。



圖 3-2-7 AfterSpeaking(Text result)

### ► BeforeSpeaking()

TextToSpeech.BeforeSpeaking : 在文字資料發出前發出的信號。



圖 3-2-8 BeforeSpeaking

## (二)、語言國家代碼

以下是您可以選擇的語言和國家代碼，編碼是先以語言別區分，再來才列出使用該語言的區域。

ces (Czech) 捷克語

CZE

BEL

IND

spa (Spanish) 西班牙語

ESP

NLD

JAM

USA

deu (German) 德語

ita (Italian) 義大利語

MHL

AUT

CHE

MLT

BEL

ITA

NAM

CHE

pol (Polish) 波蘭語

NZL

DEU

POL

PHL

LIE

eng (English) 英語

PAK

LUX

AUS

SGP

fra (French) 法語

BEL

TTO

BEL

BWA

USA

CAN

BLZ

VIR

CHE

CAN

ZAF

FRA

GBR

LUX

HKG

nld (Dutch) 荷蘭語

IRL

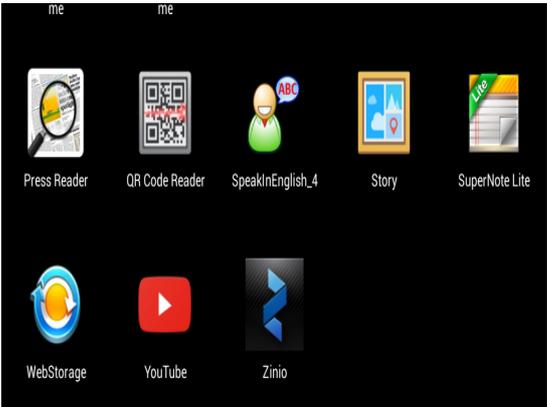
## 三、專題製作

表 3-3-1 高英高級工商職業學校教師專題製作計畫書

科別姓名	資訊 科 簡琨祥 老師
製作主題	個人英文單字學習系統
研究方法	<input type="checkbox"/> 問卷法 <input type="checkbox"/> 訪問法 <input type="checkbox"/> 觀察法 <input type="checkbox"/> 文獻蒐集 <input checked="" type="checkbox"/> 其他 ( 研究實作法 )
研究大綱	1. 如何使用 APP Inventor 2。 2. 平板電腦的深入使用。 3. APP 程式的撰寫。 4. 理解翻譯程式的運作原理。
預期效果	1. 讓英文單字的學習效果增加。 2. 隨時隨地可以學習英文。 3. 只需使用智慧型手機也能查詢英文單字。 4. 可以客製化單字內容或註解。
其 他	

#### 四、製作成果

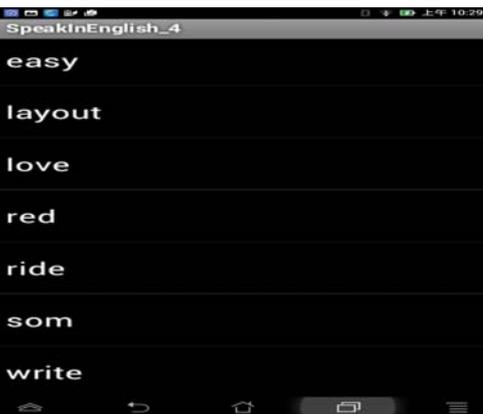
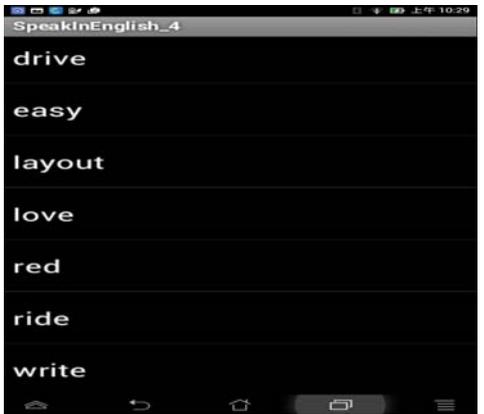
##### (一)、APP 之 SpeakInEnglish\_4 單字練習模式

	
<p>圖4-1-1 SpeakInEnglish_4啟動</p>	<p>圖 4-1-2 SIE_4 軟體初始畫面</p>
	
<p>圖4-1-3 SIE_4單字庫</p>	<p>圖 4-1-4 SIE_4 單字練習_選單字</p>
	
<p>圖 4-1-5 SIE_4 單字練習_字義</p>	<p>圖 4-1-6 SIE_4 單字練習_發音</p>

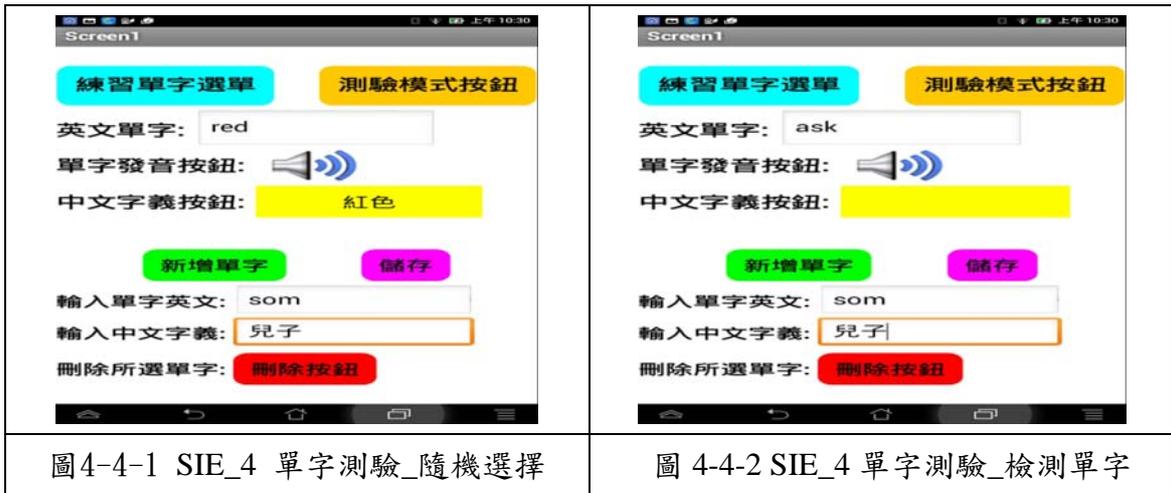
(二)、APP 之 SpeakInEnglish\_4 新增單字模式

	
<p>圖4-2-1 SIE_4 新增單字_輸入</p>	<p>圖 4-2-2 SIE_4 新增單字_儲存</p>

(三)、APP 之 SpeakInEnglish\_4 刪除單字模式

	
<p>圖4-3-1 SIE_4 刪除單字_選擇</p>	<p>圖 4-3-2 SIE_4 刪除單字_確定</p>
	
<p>圖 4-3-3 SIE_4 刪除單字_刪除鈕</p>	<p>圖 4-3-4 SIE_4 刪除單字_檢查字庫</p>

(四)、APP 之 SpeakInEnglish\_4 單字測驗模式



## 肆、製作結論與建議

### 一、結論

在此次研究中，運用了許多英文知識以及 APP 技術是我在這幾年所學習的，然而 APP 這方面的知識以及我對 APP 這方面的認知，然而我運用了現代人必須學習的英文與我擁有的專業知識讓英文有了不一樣的學習方式。

(一)一開始在製作時遇到了問題有很多如：

1. 找尋不到適合的平台

➤方法：直接結合於每個人最貼身的電子產品[手機]。

2. APP 的排版的位置

➤方法：簡潔有力不複雜。

(二)然而排除了所有遇到的困難點後，讓後續的工作都很順利，其功能顯如：

1. 可自行的增加單字。

2. 語音撥放功能正常。

3. 也可以離線使用。

### 二、建議

(一) APP 翻譯系統雖然已經大致完成，但是單字的整合功能缺乏，將來我們希望可以規劃單字類別，比如針對日常用語或是商業用語給予歸類，或是以 A-Z 的字首來加以分類，甚至是依照使用者的需求加以分類。

(二) 在製作翻譯系統的時候，有想到許多的想法，我們希望可以增加語音輸入的功能，必須能夠語音輸入及語音撥放的功能，使使用者可以校正自己的發音，並增加自己的英文口述能力。

(三) 未來希望可以將產品與書籍廠商做結合，結合推出外語組合包使得時間不足的使用者可以在最短的時間內擴充自己的單字內容，最後希望連句型文法的使用方式都可以做統合，讓智慧型手機帶來更多的教育意義。

## 參考文獻

1. 鄧文淵，2014，手機應用程式設計超簡單—App Inventor 2 初學特訓班，臺北市：碁峰出版社。
2. 鄧文淵，2014，手機應用程式設計超簡單—App Inventor 2 專題特訓班，臺北市：碁峰出版社。
3. 蔡宜坦，2014，圖解！App Inventor 2 手機程式設計教本，臺北市：旗標出版社。
4. 王安邦，2014，MIT App Inventor 2 易學易用—開發 Android 應用程式，臺北市：上奇資訊。
5. 白乃遠、曾奕霖，2014，正確學會 App Inventor 2 的 16 堂課，臺北市：博碩資訊。
6. 維基百科\_Android，取自網址 <http://zh.wikipedia.org/wiki/Android>。
7. 維基百科\_APP Inventor for Android，取自網址 [http://en.wikipedia.org/wiki/App\\_Inventor\\_for\\_Android](http://en.wikipedia.org/wiki/App_Inventor_for_Android)
8. App Inventor 2，取自網址 [ai2.appinventor.mit.edu/](http://ai2.appinventor.mit.edu/)
9. APP Inventor 入門準備，取自網址 <http://s810802.pixnet.net/blog/post/47679122-%E3%80%90mit-app-inventor85%A5-%E5%96%80%E6%BA%96%E5%82%99%E3%80%91>
10. APP Inventor 中文學習網，取自網址 <http://www.appinventor.tw/filepool>
11. MIT App Inventor，取自網址 [appinventor.mit.edu/](http://appinventor.mit.edu/)