

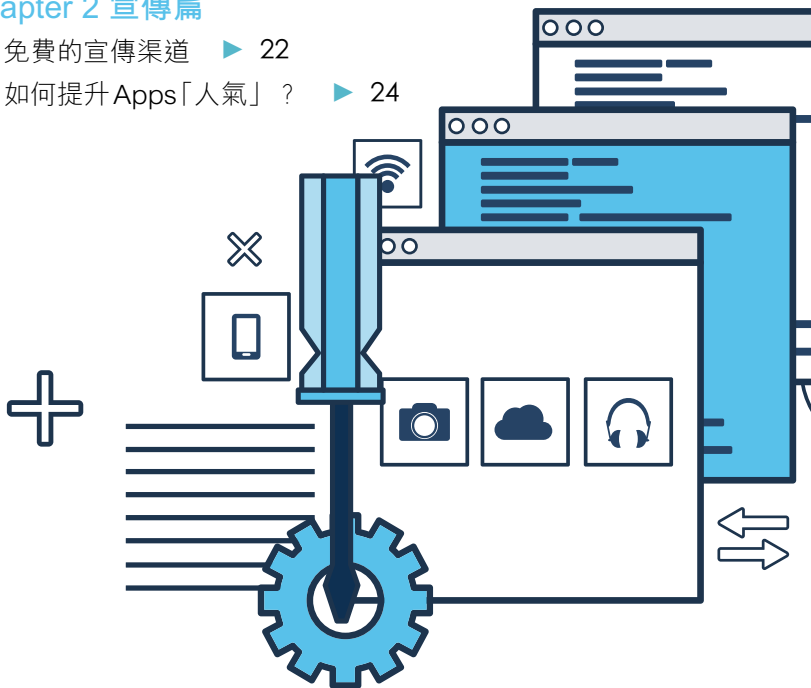
前言

Chapter 1 構思篇

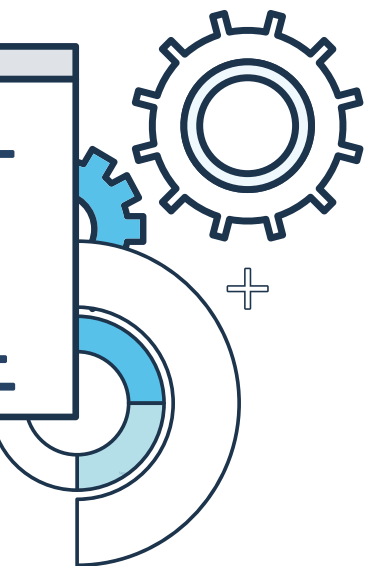
- 1.1 創作意念從何而來？ ▶ 8
- 1.2 我的構思可以實現嗎？ ▶ 10
- 1.3 構思一定天馬行空？ ▶ 11
- 1.4 怎樣的介面才可吸引用家？ ▶ 13
- 1.5 介面設計是否 User-friendly？ ▶ 16

Chapter 2 宣傳篇

- 2.1 免費的宣傳渠道 ▶ 22
- 2.2 如何提升 Apps「人氣」？ ▶ 24



目錄



Chapter 3 集資篇

- 3.1 如何令 Apps 有利可圖？ ▶ 28
- 3.2 籌募資金的渠道 ▶ 30
- 3.3 何謂天使投資？ ▶ 32

Chapter 4 工具篇

- 4.1 介紹 App Inventor ▶ 36
- 4.2 開啟 New Project ▶ 41
- 4.3 將裝置與 App Inventor 連接 ▶ 42
- 4.4 Android Basic Competent ▶ 47
- 4.5 Android Layout Competent ▶ 95
- 4.6 Android Media Competent ▶ 100
- 4.7 Sensors ▶ 115
- 4.8 Social ▶ 119
- 4.9 Storage ▶ 123
- 4.10 Connectivity ▶ 132

- Appendix : 拼圖解說 ▶ 156

Chapter 4

工具篇



Chapter 4：工具篇

「工欲善其事，必先利其器」，對App開發者來說，無論有多精采的Idea，如果不能化為現實，一切也是枉然，所以在這個章節中，我們將會先就一些較常用的開發工具作介紹，坊間現有不同的開發流動應用程式(Apps)的工具，如開發iOS的XCode，Android的ADT Bundle及Window Phone OS的Visual Studio等。這些開發工具的功能都很齊備，然而卻要求開發者須具有較豐富的程式開發經驗及知識，而且上手時間亦較長。

對於部分沒有程式開發經驗的讀者而言，筆者建議可先從開發工具美國麻省理工學院的MIT App Inventor入手，它是針對Android作業系統而設，操作較簡易，本書亦會以這款開發工具來作示範。



4.1 介紹 App Inventor

早於2010年，App Inventor已經推出測試版本予不同學校使用。App Inventor以Scratch的拼圖形式的程式開發，這個製作方法對缺乏程式開發經驗的人來說較為方便易

明，因使用者能以拖拽拼圖的方式去開發程式，完全不用任何程式碼便可完成開發，大大降低程式開發的門檻。經過多次測試及改良後，App Inventor 於2010年尾正式開放給所有使用者。

App Inventor的網址：<http://appinventor.mit.edu/explore/>

初啟流程

1. 首先進入App Inventor的首頁，然後按"Create"。



2. 然後按"New Project"及輸入專案的名稱，然後新的專案已經設定好了。





在畫面正上方有一些選單及功能，簡易說明如下：

- **Projects (專案)**
主要是用來開啟、匯入匯出項目及金鑰。
- **Connect (連線)**
功能主要有連結模擬器及智能手機用作測試。

另外要在開啓程式時載入Slider按鈕的初始位置，所以在Screen1中拉出"when Screen1.Initialize do"拼圖，Label1中拖出"set Label1.Text to"拼圖及Slider1的"Slider1.ThumbPosition"拼圖組成下圖：



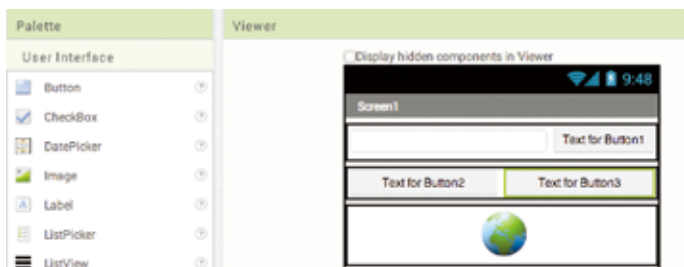
完成後可連接模擬器運行，開啟應用程式時Label1的內容為0，因為Slider的按鈕位置被設定為0。然後我們可拖拽Slider按鈕，Label1的數值亦會隨著改變。



viii. WebView

這次我們會製作一個網頁，並有三個按鈕可以控制網頁。

首先在 My Projects 頁面，建立一個名為 ex5_d_8 的專案。然後由 Palette 視窗中的 Layout 選項，拖拽兩個 HorizontalArrangement 到 Viewer 視窗，再從 User Interface 中先後拉出一個 TextBox 及一個 Button 到 HorizontalArrangement1 中，然後再拖拽兩個 Button 到 HorizontalArrangement2 中，最後拉一個 WebView 到 HorizontalArrangement2 下面，如下圖：



首先我們在 Components 視窗中點選 WebView1，然後我們在 Properties 視窗中可設定 WebView1 的屬性。屬性的詳細解釋如下：

FollowLinks：

設定 WebView 會否記錄連結。
如果設定為是，使用者可以返回
上一頁或下一頁。

HomeUrl :

設定 WebView1 的首頁。

PromptforPermission :

設定會否有權限提示彈出。如設定為否，會當成已經給予權限。

UsesLocation :

設定應用程式有沒有權限使用使用者的地理位置。



然後點選 WebView1 設定剔選 FollowLinks，HomeUrl 輸入 <http://www.google.com/>，剔選 PromptforPermission，UsesLocation 不用剔選，Width 和 Height 都設為 Fill Parent。

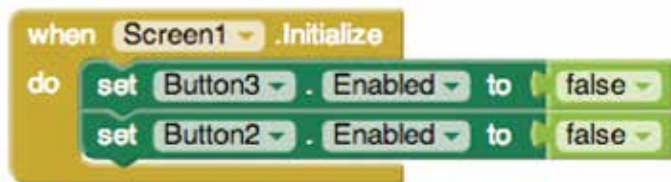
接著我們開始設定 HorizontalArrangement1 的部分，點選 HorizontalArrangement1 設定 Width 為 Fill parent，Height 為 35 pixels。再點選 TextBox1 設定 Hint 為 Enter URL 和 Width 為 Fill Parent。最後點選 Button1 並設定 Text 為 Go。

最後則設定 HorizontalArrangement2 的部分，點選 HorizontalArrangement2 設定 Width 為 Fill parent。再點選

Button2 設定 Text 為 Back 和 點選 Button3 並設定 Text 為 Next。

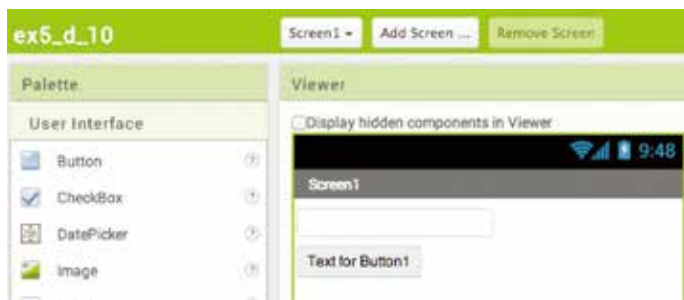
然後就開始代碼部分，首先點擊右上角的 Blocks 按鈕。當開啓程式時，使用者並不能使用上一頁或下一頁，所以要使 Button2 及 Button3 按鈕不能使用，拼圖設定是先點選 Screen1 並拖出 "when Screen1.Initialize do" 拼圖。

接著，點選 Button3 拖出 "set Button3.Enabled to" 拼圖，放置在 "when Screen1.Initialize do" 拼圖內，再點選 Button2 拖出 "set Button2.Enabled to" 拼圖，放置在 "set Button3.Enabled to" 拼圖之下，再從 "Logic" 中拖出兩個 "false" 拼圖，分別放置於 "set Button3.Enabled to" 和 "set Button2.Enabled to" 後方，完成圖如下：

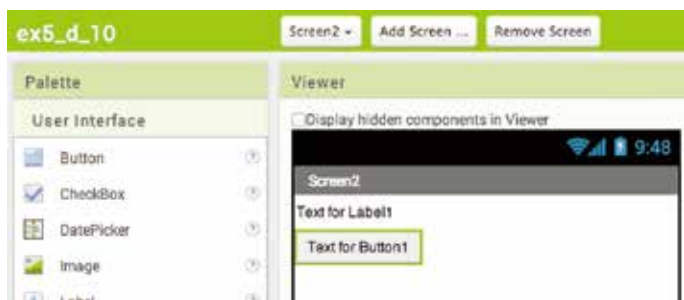


然後我們設定 Button1 按鈕，首先我們要檢查使用者輸入的 URL 有沒有包含 "http://" 或 "https://" 的字眼。如果 URL 內有 "http://" 或 "https://" 的字眼，當按下 Button1

如以下圖示：



因為這次要有兩個畫面的轉換，所以我們點擊 Add Screen 按鈕(於 Viewer 視窗上方位置)，然後新增一個新的畫面 "Screen2"，後我們可點擊左手的 "Screen1" 按鈕並選擇 "Screen2"。這時我們可以看到 Viewer 視窗中已換成新的畫面 "Screen2"。我們拖拽一個 Label 及一個 Button 到 Screen2，如下圖：



然後我們要將輸入在 TextBox 中的內容傳送到 Screen2 的 Label，所以要在 Blocks 中以代碼完成。在 Block 工作區中先揀選 Screen1 視窗，然後拉出 Button 的 "when Button1.Click do"，在內放置 Control 的 "open another screen with start value screenName startValue" 拼圖，上方空格放置 Text 的 " "，並輸入 "Screen2"，下方空格放置 TextBox1 的 "TextBox1.Text"，組成以下拼圖：



接著，揀選 Screen2 的視窗（上圖箭嘴），這時 Viewer 視窗中應該沒有任何拼圖。在 Screen2 我們要在進入時載入 Screen1 傳遞的數值，及使用者可以透過點擊按鈕返回上一個畫面。我們首先處理載入的部分，拉出 Label1 的 "set Label1.Text to"，放置到 Screen2 的 "when Screen2.Initialize do" 內，再在後方放置 "Control" 中的 "get start

value" 拼圖；接著，我們處理返回上一個畫面的部分，拉出 Button1 的 "when Button1.Click do" 及 Control 的 "close screen" 並組成第二塊拼圖，可參考以下拼圖：



最後運行模擬器進行測試，緊記要點選 Screen1 才可以開始運行，否則一開始顯示的就會是 Screen2。首先於 TextBox 中輸入內容，然後點擊按鈕進入 Screen2。在 Screen2 我們可以見到 Label 的內容變為我們輸入的內容，最後使用者可按按鈕返回 Screen1。程式截圖如下：





4.5 Android Layout Competent

在這個部分我們將會研究畫面布局的方法，大致上可以分成三種：Horizontal Arrangement（橫向布局）、Vertical Arrangement（直向布局）和Table Arrangement（並向布局）。

每種布局都有不同的特性，使用不同的布局有助於不同的表達，以下將會逐一解釋不同布局的用法。

i. Horizontal Arrangement & Vertical Arrangement

Horizontal Arrangement 的特性就是將畫面元件橫向並排，而Vertical Arrangement的特性就是將畫面元件上下並排。這次我們會製作一個混合使用Vertical Arrangement及Horizontal Arrangement的應用程式。

首先在My Projects頁面，建立一個名為ex5_d_11的專案。

先點選Components中的Screen1並設定Scrollable為否，否則畫面元件的Height設定為Fill Parent，也