

用MIT App Inventor2程式拼圖來開發Android裝置應用程式

李啟龍

學習 Android 裝置程式設計，可以不必學習較為艱澀的 Java 語法，只要使用拼圖模式來組程式，就可以完成 Android 裝置的應用程式。MIT App Inventor 2 網站提供雲端服務，讓我們可以免費使用其提供的強大功能，來設計 Android 裝置上的 App 應用程式，完成作品後還可以上傳到 Google Play 商店，提供給大眾下載使用喔。

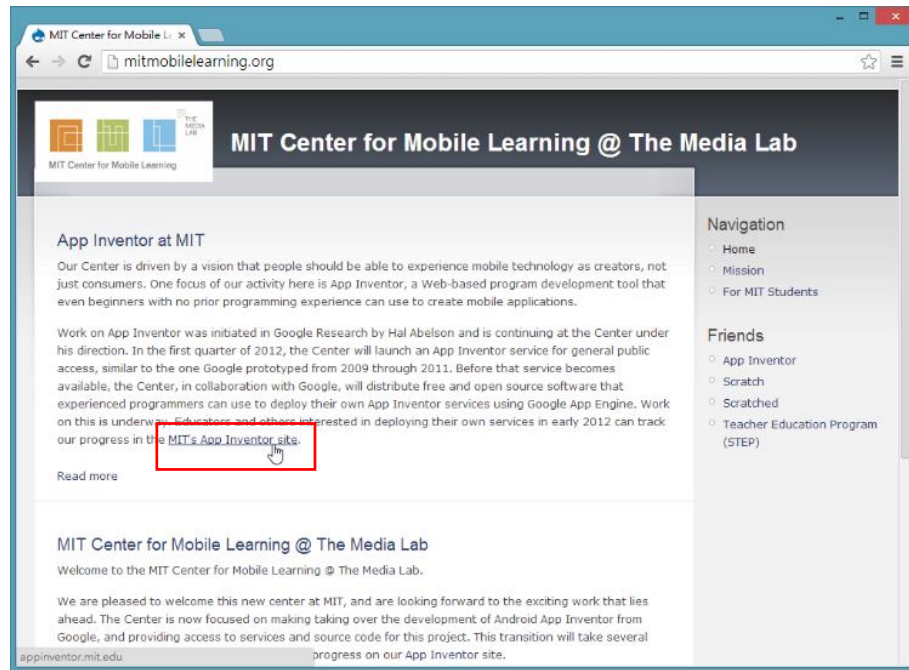
本月份的電子報，我們就來進行用 MIT App Inventor2 程式拼圖，來開發 Android 裝置 App 程式的相關說明：

➤ MIT App Inventor 2 簡介

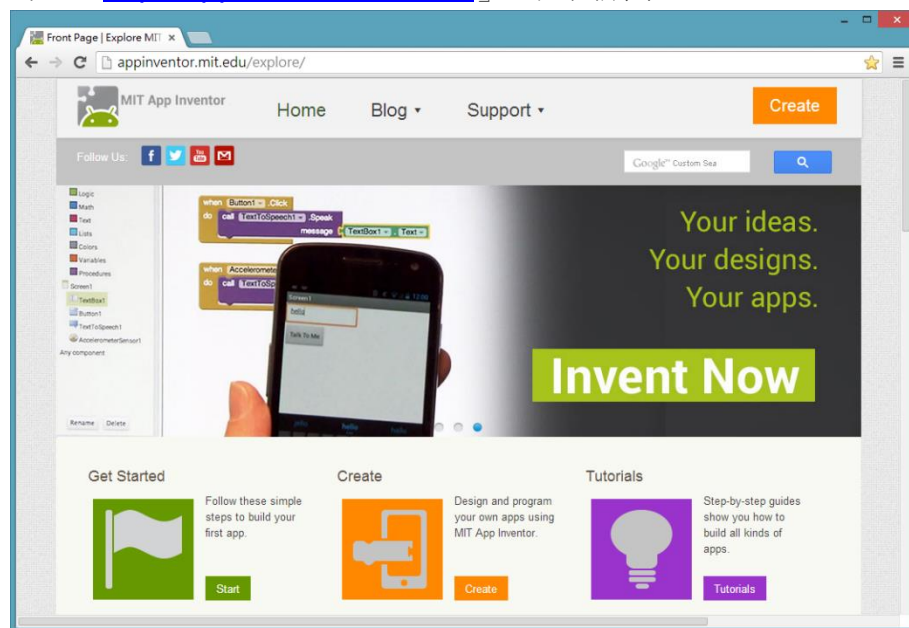
Android 作業系統是一個以 Linux 為基礎的開放原始碼作業系統，中文俗稱為「安卓」，其起源最早是在西元 2003 年由安迪魯賓（Andy Rubin）所開發製作，目前由 Google 公司成立的開放手持設備聯盟 Open Handset Alliance(OHA) 來持續開發主導。

Android 作業系統的應用層面，最早是由智慧型手機開始，後來逐漸擴展到平板電腦及其他領域上，在西元 2010 年底，Android 作業系統成為全球第一大智慧型手機作業系統，其市佔率超過 Apple 公司的 iOS 作業系統。

Google 公司在西元 2010 年，為了讓更多人在沒有程式基礎的狀況下，能夠開發 Android 裝置應用程式，推出 Google App Inventor for Android 工具軟體，讓使用者可以運用拼圖模式來組程式語法，進而發展 Android 裝置的應用程式。後來，Google 公司在西元 2012 年將此軟體，移轉給美國麻省理工學院（MIT）的行動學習中心（MIT Center for Mobile Learning）來維護，行動學習中心的網址為：「<http://mitmobilelearning.org/>」，其網站如圖所示。點選其中的「MIT's App Inventor site」超連結，會連結到 MIT App Inventor 2 網站。



麻省理工學院是在西元 2013 年 12 月 3 日推出 MIT App Inventor 2 網站，提供免費的雲端服務，使用者可以透過瀏覽器來開發 Android 裝置應用程式，該網站的網址為：「<http://appinventor.mit.edu/>」，如圖所示。



網站上提供了許多詳細的教學資源，來引導使用者進入 App 應用程式開發的世界，使用者開發完成的程式，可以下載到任何使用 Android 作業系統的裝置，包括：智慧型手機或平板電腦。

在 App 程式開發過程中，我們可以使用模擬器或 Android 裝置，來測試應用程式執行的狀況，待完成相關開發作業後，更可以將應用程式，上傳到 Google Play 商店提供大眾下載安裝。

➤ 建置 App Inventor 2 開發環境

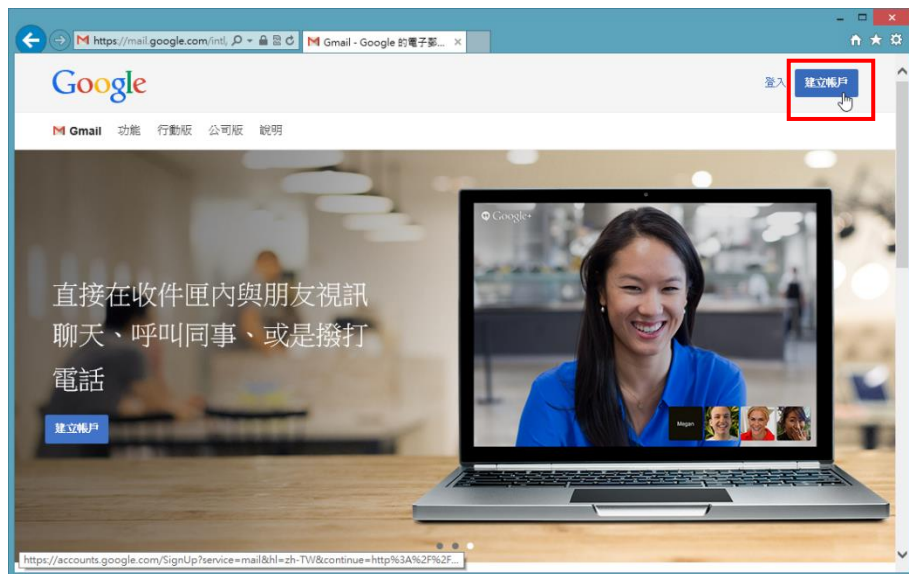
使用 MIT App Inventor 2 來開發 Android 程式，其開發動作都是在瀏覽器上完成，為網路雲端開發環境，要建置 MIT App Inventor 2 的開發環境，需要進行 3 項前置作業：

1. 申請 Gmail 帳號。
2. 安裝並測試 Java 環境。
3. 安裝 MIT App Inventor 2 軟體（為了以模擬器或 USB 連線來測試程式）。

1.申請 Gmail 帳號

要使用 MIT App Inventor 2 開發 Android 程式，需要擁有 Gmail 帳號，藉此來登入開發網站，申請 Gmail 帳號的步驟如下：

Step1：在瀏覽器的網址列輸入網址：<http://mail.google.com>，到達 Gmail 信箱的登入頁面，接著點選「建立帳戶」按鈕，如圖所示。



Step2：接著輸入要建立的 Google 帳戶資料，包括：「名稱」、「使用者名稱」、「密碼」、「生日」、「性別」...等資料，如圖所示。Google 將所有的產品與服務全部整合到一個帳戶，只要使用一組使用者名稱與密碼登入，就能使用各種豐富的 Google 服務。

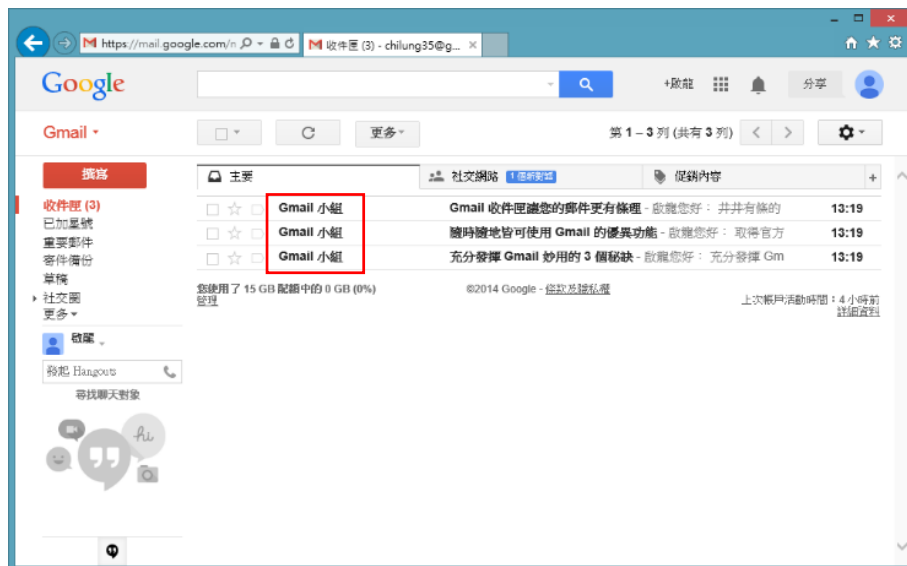
填寫資料時，需將「我同意 Google《服務條款》及《隱私權政策》」的核取方塊打勾，才能按下「下一步」按鈕，如圖所示。

Step3：設定想要顯示的資訊，此部分的設定就依個人的使用需求來輸入，此階段為讓使用者上傳大頭照，如圖所示。照片上傳後，以後仍然隨時可更換相片。

Step4：完成 Gmail 信箱的申請，按下「前往 Gmail」按鈕，即可開始使用 Google 帳戶，如圖所示。



第一次成功登入 Gmail 信箱的主畫面如下，信箱中會收到 Gmail 小組寄來的信件。



2.安裝並測試 Java 環境

要成功執行 MIT App Inventor 2 程式，在電腦中需要建置 Java 環境。由於 Java 的使用非常普及，所以在電腦中往往已經有安裝過 Java 環境，我們先來測試一下，目前電腦是否已經完成 Java 的安裝，測試步驟如下：

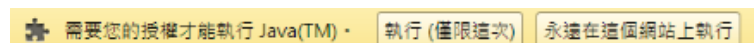
Step1：在瀏覽器的網址列輸入網址：「<http://www.java.com/>」，連結到 Java 首頁後，按下其中的「我有 Java 嗎？」超連結，便會連結到測試網頁。



當然也可以在瀏覽器直接輸入測試 Java 環境的網址：
「http://www.java.com/zh_TW/download/installed.jsp」，點選其中的「驗證 Java 版本」按鈕，以進行 Java 版本的測試，如圖所示。



Step2：在驗證 Java 版本的過程中，由於瀏覽器的安全性設定，也許會出現授權提示，如圖所示，此時我們要點選「執行（僅限這次）」或「永遠在這個網站上執行」都可以。



Step3：接著出現是否執行應用程式的提示視窗，此處我們選擇「執行」按鈕，以進行 Java 版本的驗證。



Step4：最後出現 Java 版本驗證完成的網頁。MIT App Inventor 2 需要在具有 Java 6 以上的電腦環境中執行，請讀者注意電腦的 Java 版本是否為 Java 6 以上。



如果電腦中並未安裝 Java 6 以上的 Java 版本，我們就需要到 Java 的首頁進行免費 Java 的下載，其安裝步驟如下：

Step1：在瀏覽器的網址列輸入網址：「<http://www.java.com/>」，連結到 Java 首頁後，按下其中的「免費 Java 下載」按鈕，便會連結到下載網頁。



Step2：此處出現建議下載的 Java 版本，選擇「同意並開始免費下載」按鈕，便會下載可安裝的 Java 程式，執行該程式便可開始安裝 Java，如圖所示。如果遇到安全性警告，需要選擇「執行」按鈕，才能繼續安裝。



Step3：接著出現歡迎使用 Java 視窗，此處建議使用預設安裝設定，所以直接按下「安裝」按鈕即可，如圖所示。



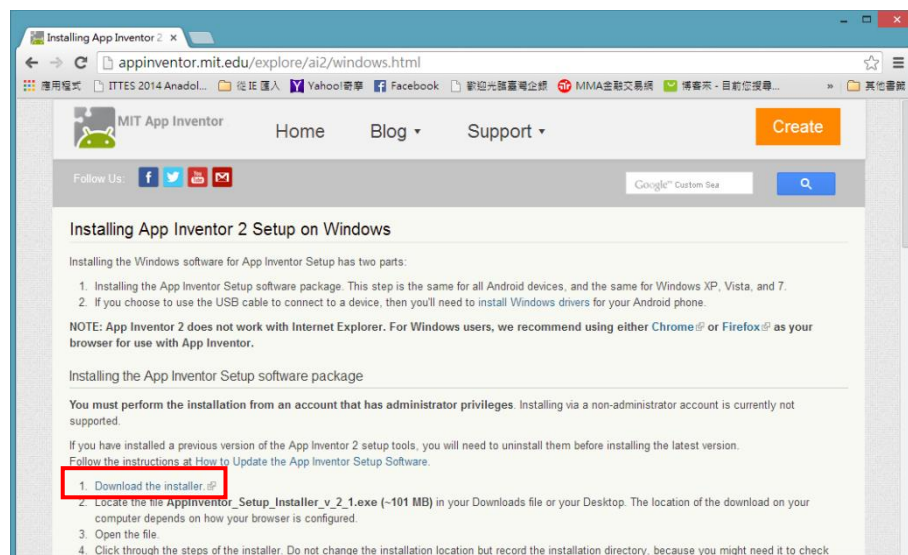
Step4：最後出現成功安裝 Java 畫面，此時按下「關閉」按鈕即可。



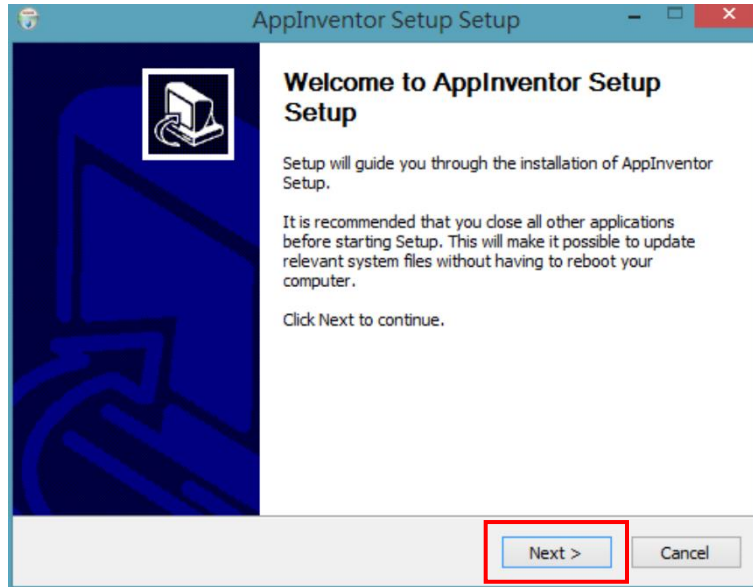
3. 安裝 MIT App Inventor 2 軟體

完成了 Gmail 帳號的申請以及建置了 Java 6 以上版本的 Java 環境之後，接下來就是安裝 MIT App Inventor 2 軟體，以便於在沒有 Android 裝置的情況下，以模擬器來測試 Android 程式。安裝 MIT App Inventor 2 軟體的步驟如下：

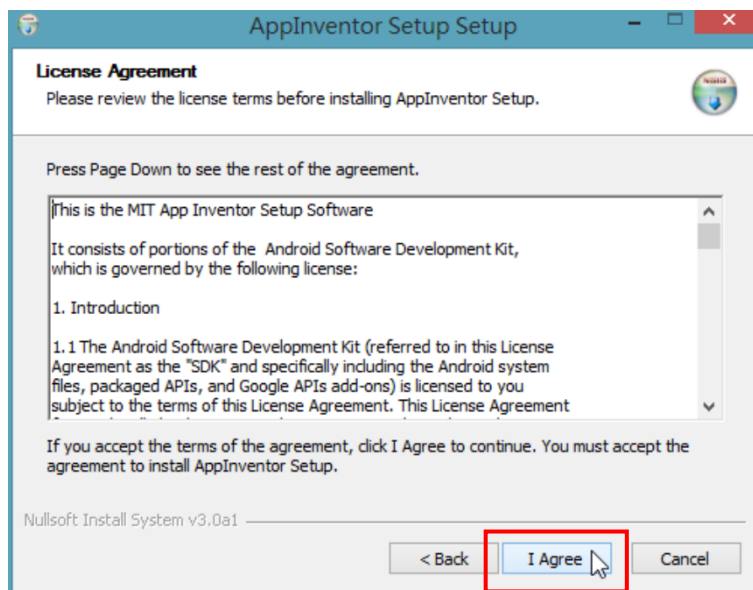
Step1：在瀏覽器的網址列輸入網址：
「<http://appinventor.mit.edu/explore/ai2/windows.html>」，到達 MIT App Inventor 2 的官方下載網頁，點選其中的「Download the installer」超連結，如圖所示。



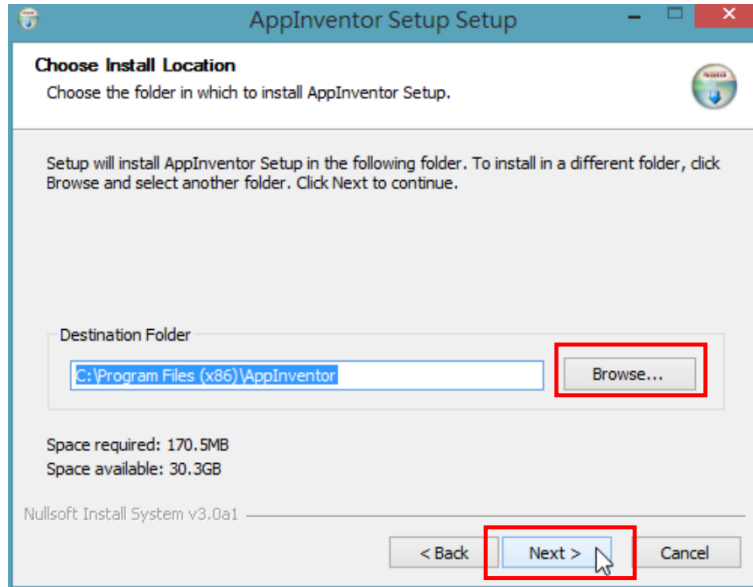
Step2：下載的軟體名稱為：「AppInventor_Setup_Installer_v_2_2.exe」，緊接著執行該軟體，會出現歡迎安裝的畫面，按下「Next」按鈕以進行軟體安裝。本書光碟也有提供 AppInventor_Setup_Installer_v_2_2.exe 軟體（約 92MB），可以直接安裝使用。



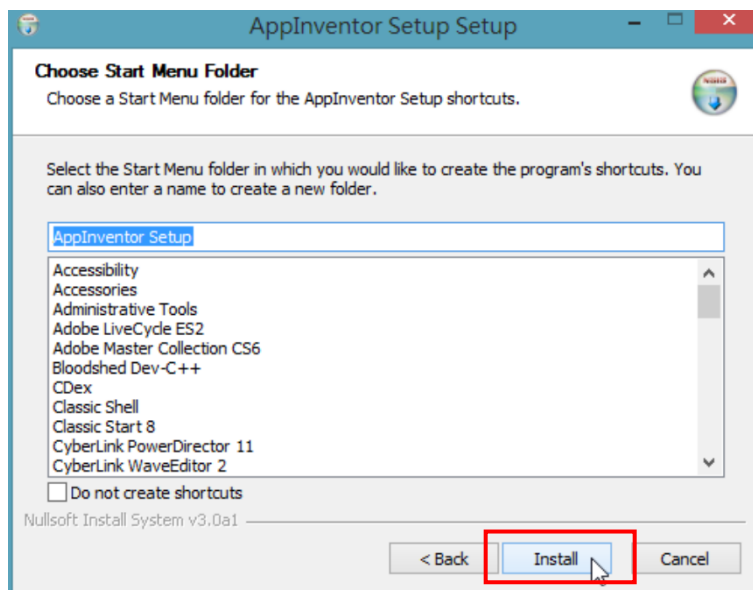
Step3：必須同意相關的版權條款，才能繼續進行安裝使用，此處選擇「I Agree」按鈕，以繼續進行軟體安裝。



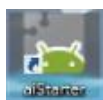
Step4：此處要選擇軟體的安裝路徑，各位讀者可以依據自己的需求按下「Browse...」按鈕更換路徑；如不更換安裝路徑，按下「Next」按鈕以繼續安裝。



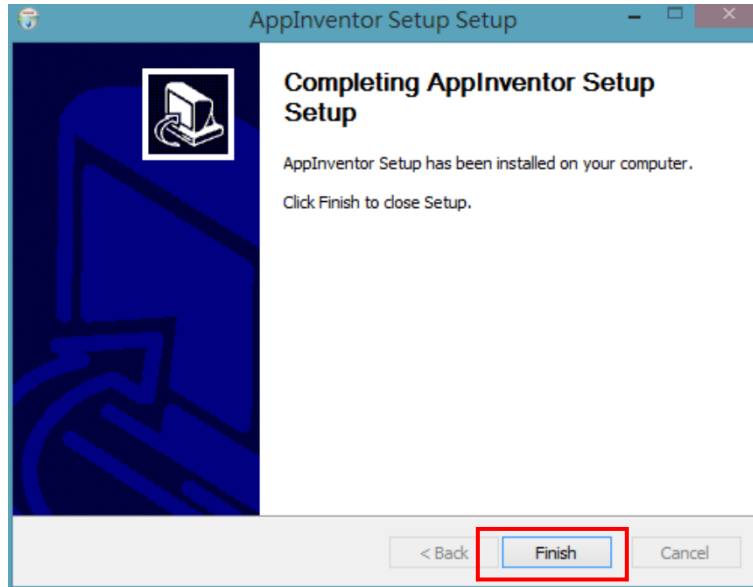
Step5：選擇在開始功能表中的資料夾名稱，建議依預設值安裝，按下「Install」按鈕以進行安裝。



Step6：順利完成程式的安裝，按下「Finish」按鈕以結束 MIT App Inventor 2 軟體的安裝，在程式集裡會多了「aiStarter.exe」程式 ，桌面也會多了



圖示。



➤ 我的第一支 Android 程式

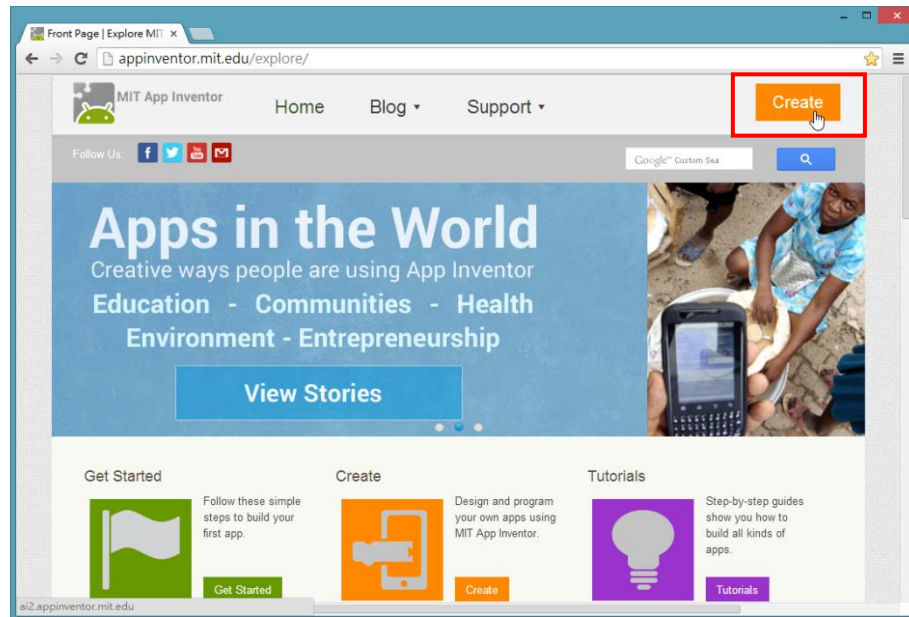
在完成了 MIT App Inventor 2 開發環境的建置之後，現在我們將以一個簡單的例子，帶領讀者來撰寫第一支 Android 程式。

使用 MIT App Inventor 2 來開發 Android 裝置的應用程式，其整體設計的概念可分為 3 個階段，說明如下：

- 階段 1：使用 Design 畫面來設計 Android 裝置的使用者介面。
- 階段 2：使用 Blocks Editor 畫面來設計程式的控制邏輯。
- 階段 3：使用 Android 裝置或模擬器來測試程式的執行。

開發我的第一支 Android 程式的步驟如下：

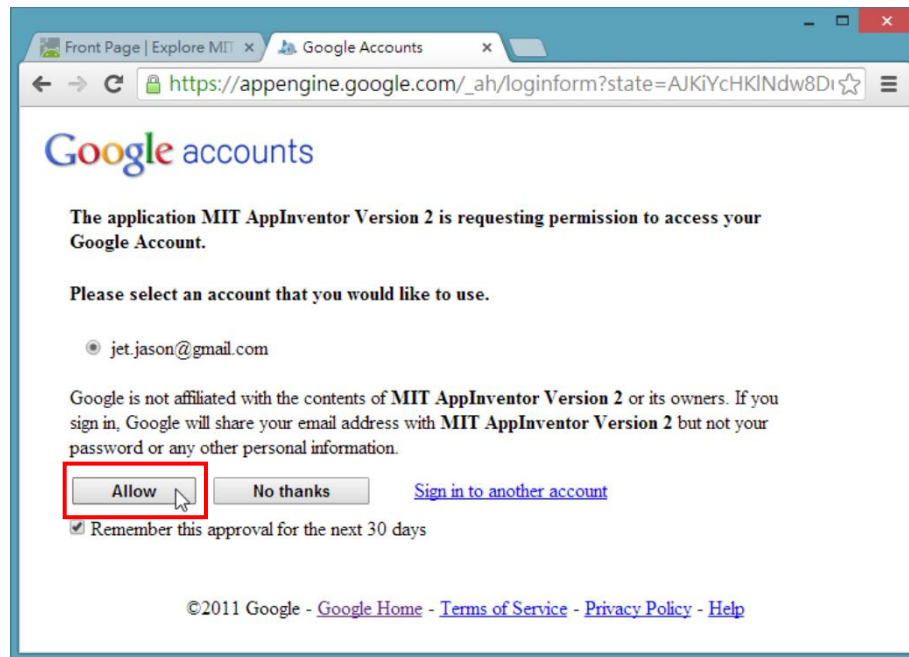
Step1：在瀏覽器的網址列輸入網址：「<http://appinventor.mit.edu/explore/>」，到達 MIT App Inventor 2 網站，按下右上角的「Create」按鈕，以建立 Android 裝置的應用程式。



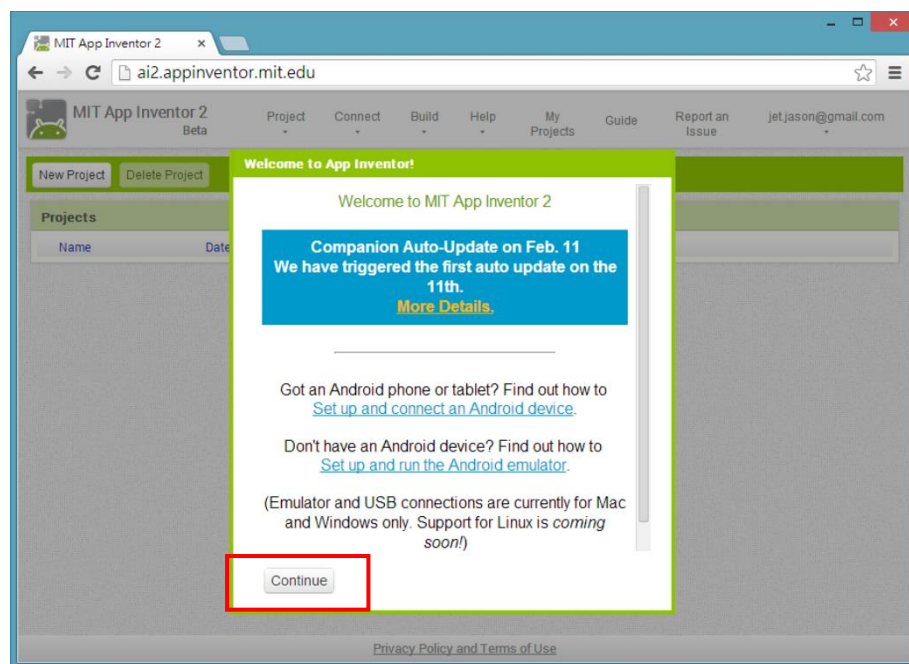
Step2：第一次使用 MIT App Inventor 網站開發 Android 裝置程式，需要登入 Google 帳戶，請各位讀者輸入自己的 Gmail 信箱與密碼，然後按下「登入」按鈕，如圖所示。



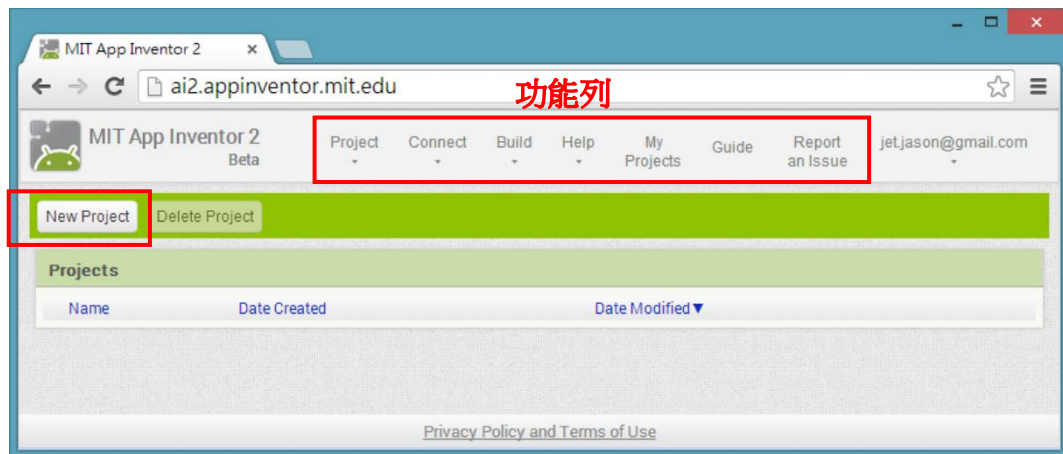
Step3：接著詢問是否允許 MIT App Inventor 2 存取這個 Google 帳戶，確定是這個帳戶就按下「Allow」按鈕，如圖所示。



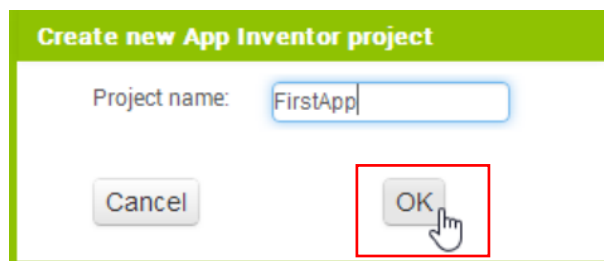
Step4：成功登入 MIT App Inventor 2 網站，出現歡迎訊息，按下「Continue」按鈕，開始準備進行新專案的設計。



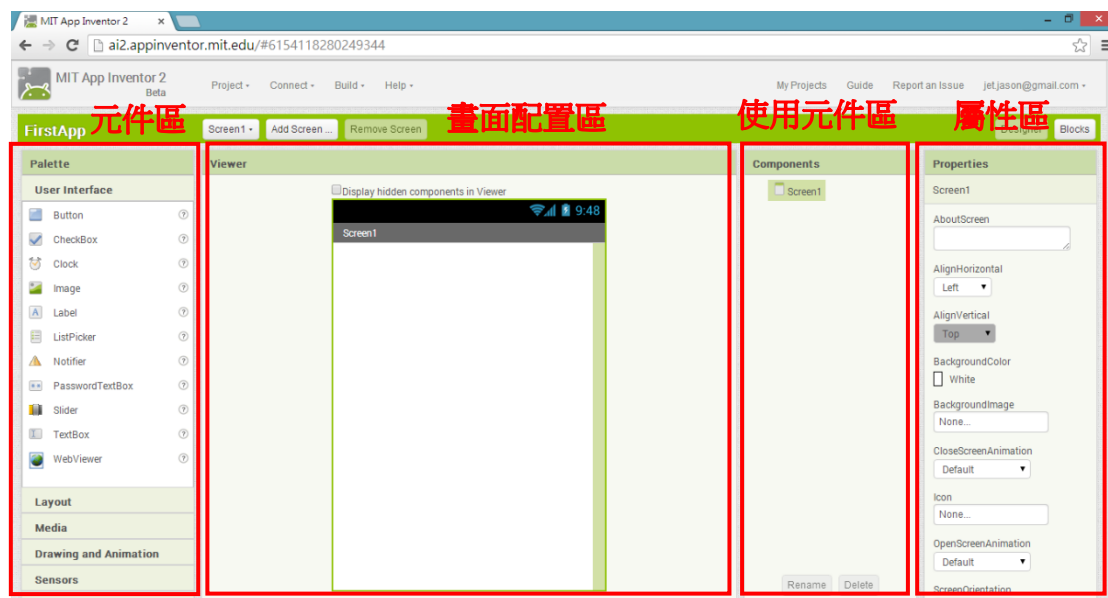
Step5：MIT App Inventor 2 網站的專案管理畫面如下圖所示，最上方是「功能列」。點選「New Project」按鈕來新增一個專案。



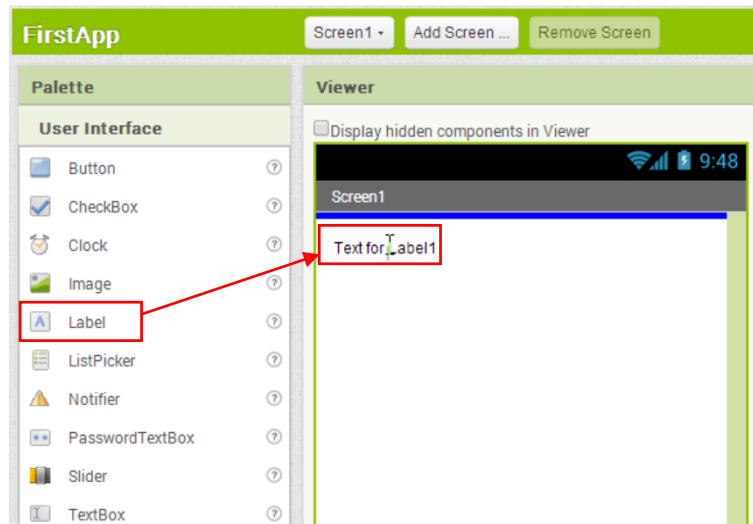
Step6：輸入新的專案名稱，此處我們輸入「FirstApp」當作第一支 Android 程式的名稱，然後按下「OK」按鈕來確認新增專案名稱。



Step7：進入「FirstApp」專案的設計頁面，專案名稱「FirstApp」位於左上角，此頁面包含 4 個區塊，各區塊名稱如下圖所示，由左至右依序為「元件區」、「畫面配置區」、「使用元件區」、「屬性區」。

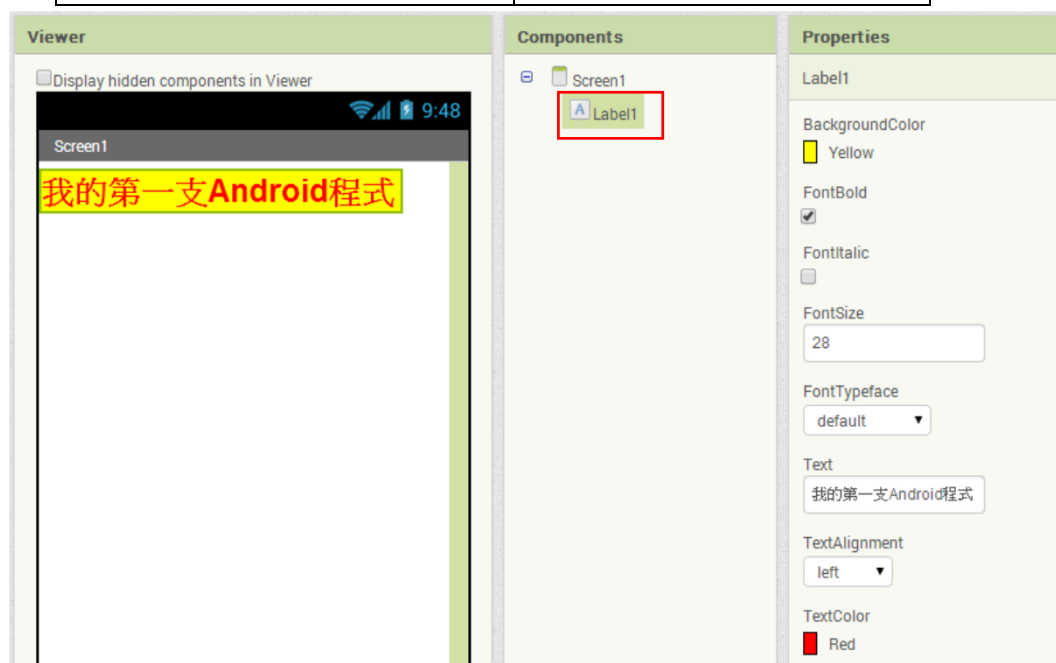


Step8：從「元件區」中用滑鼠左鍵拖曳一個 Label 元件進入「畫面配置區」，Label 元件的中文名稱為標籤，主要是用於顯示文字，如圖所示。



Step9：拖曳完 Label 元件之後，在「使用元件區」中會看到一個 Label1 元件，點選 Label1 元件後，可以在「屬性區」中，進行 Label1 元件的屬性值設定。如圖將 Label1 元件的屬性值設定如下：

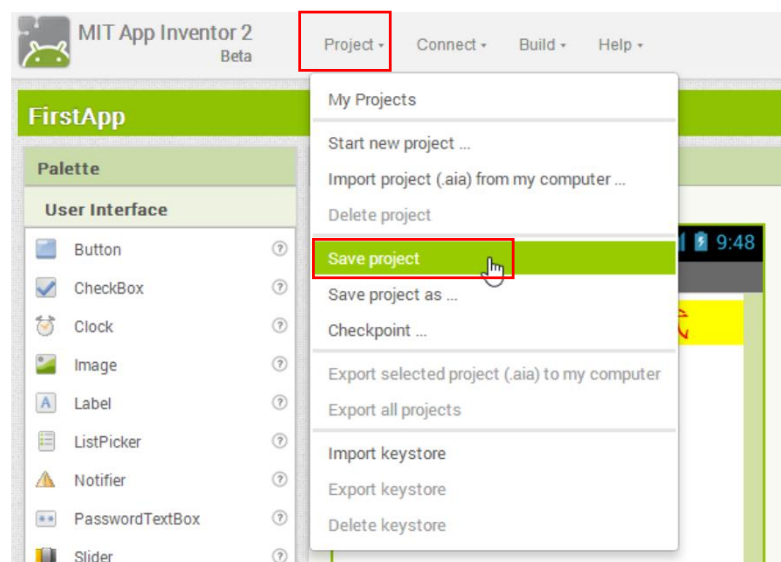
屬性	屬性值
BackgroundColor（背景顏色）	Yellow
FontBold（文字粗體）	勾選
FontSize（文字大小）	28
Text（顯示文字）	我的第一支 Android 程式
TextColor（文字顏色）	Red



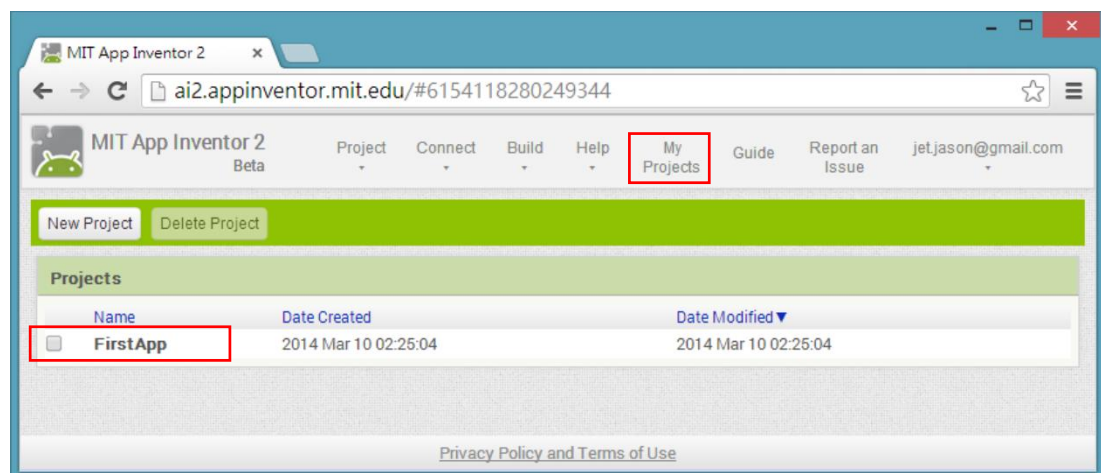
Step10：剛剛設定完成的 Label1 標籤，其寬度比裝置的螢幕寬度短，如果想要讓 Label1 的寬度與螢幕同寬，我們可以將 Label1 元件的「Width」屬性設為「Fill parent」，然後按下「OK」按鈕，其結果如圖所示，我們可以觀察到 Label1 元件的背景顏色寬度與裝置螢幕寬度相同。



Step11：此時我們已經完成了一個非常簡單的 Android 程式，可以在行動裝置上顯示一段文字，文字內容為「我的第一支 Android 程式」。選取執行功能列上的【Project/Save project】選項，可以將此專案儲存下來。



Step12：選取執行功能列上的【My Projects】選項，可以發現我們的第一個 FirstApp 專案已經被儲存下來了，未來如需重新設計編輯該專案，只要點選專案名稱即可。



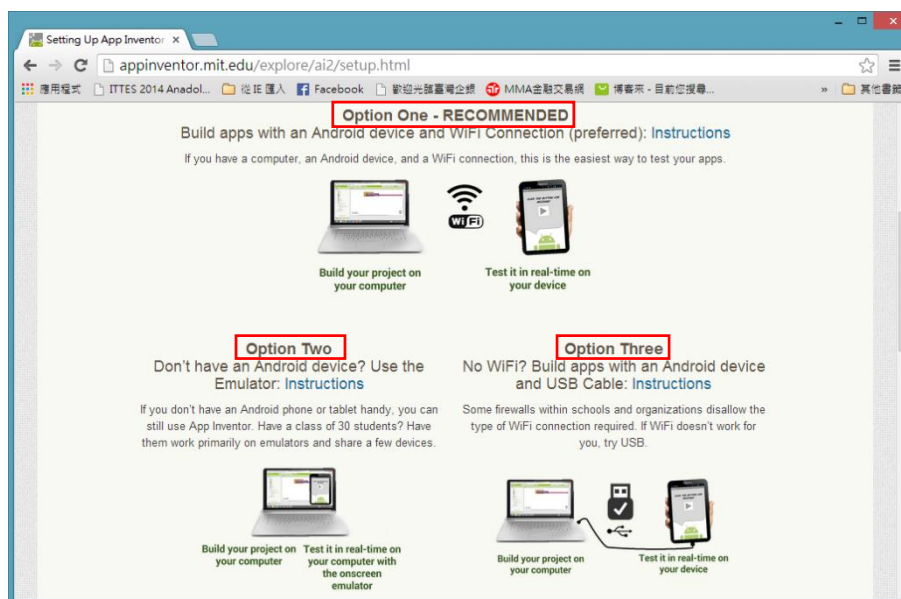
小提示：

如果要連到 MIT App Inventor 2 的開發頁面，我們可以直接在瀏覽器的網址列輸入：「<http://ai2.appinventor.mit.edu/>」。至於瀏覽器的選擇，當然是優先推薦 Google Chrome 啦！

➤ 測試執行開發好的程式

我們要測試執行開發完成的程式（Live Testing），依據 MIT App Inventor 2 官方網站的說明，網址為：「<http://appinventor.mit.edu/explore/ai2/setup.html>」，主要有三種選擇方式，說明如下：

- 第 1 種方式：適用於已經有 Android 裝置，電腦與裝置之間有 WiFi 連線，此方式是測試執行 App 程式最容易的方式。
- 第 2 種方式：適用於沒有 Android 裝置，使用模擬器（Emulator）來測試執行 App 程式。
- 第 3 種方式：適用於已經有 Android 裝置，但是電腦與裝置之間無法使用 WiFi 連線，此方式是使用 USB 連線來測試執行 App 程式。



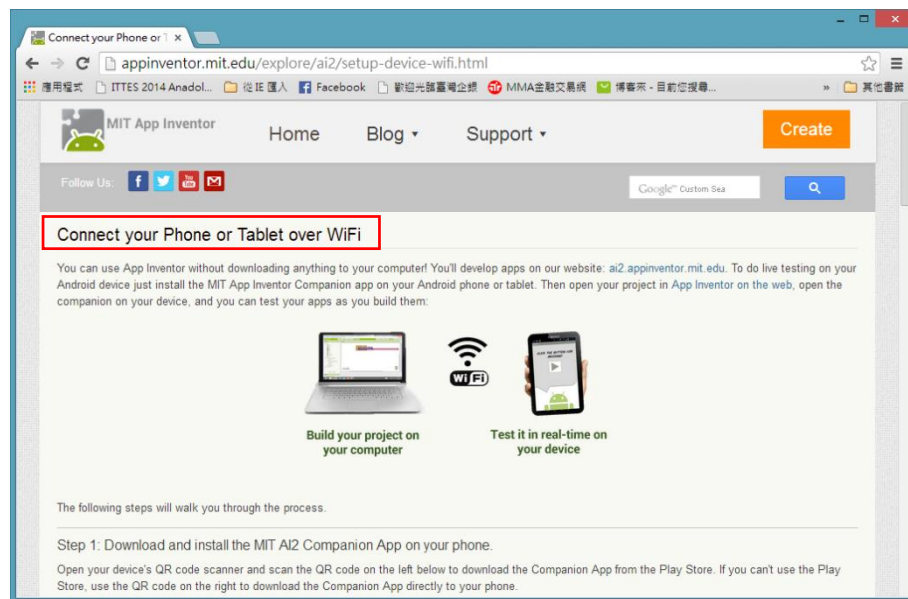
以下分別說明 3 種測試執行 App 程式的方式。

1. WiFi 連線方式

使用 WiFi 連線方式並不會把要測試執行的程式，安裝到 Android 裝置上，此方式是使用 Android 裝置來執行程式並顯示結果，當連線中斷時，Android 裝置上的程式就會消失。

依據 MIT App Inventor 2 官方網站的說明，WiFi 連線方式的說明網址為：

「<http://appinventor.mit.edu/explore/ai2/setup-device-wifi.html>」，如圖所示。



設定 WiFi 連線方式的步驟如下：

Step1：下載並安裝 MIT AI2 Companion App 到 Android 裝置上。我們可以使用 Android 裝置的 QR code 掃描程式（例如：Play 商店中的 QR Droid 程式，可自 Play 商店中免費下載使用），掃描下圖的 QR code 來下載 MIT AI2 Companion App 程式，官方網頁的建議是採用 Play 商店下載並安裝的方式。

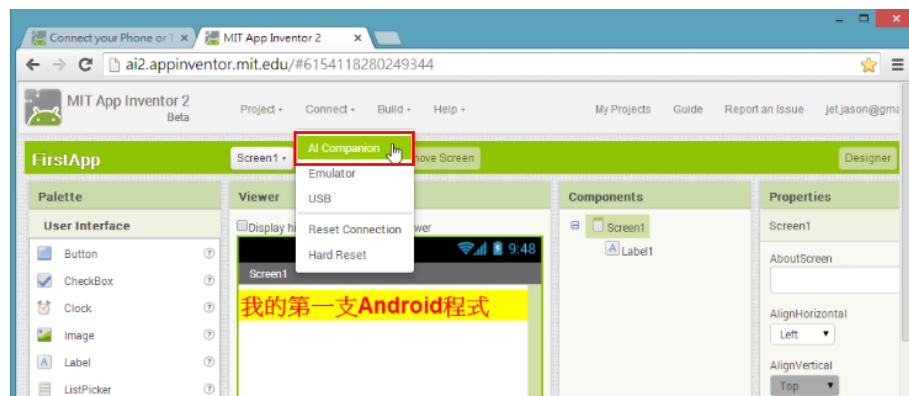


安裝完成後，會在 Android 裝置上，看到 MIT AI2 Companion App 的圖示，如圖所示。

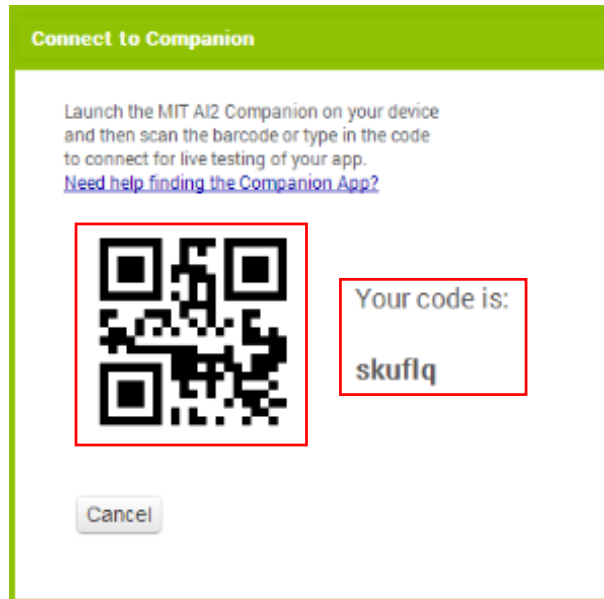


Step 2: 將電腦與 Android 裝置連接到同一個 WiFi 網路基地台，以藉由 WiFi 網路來傳送資料。

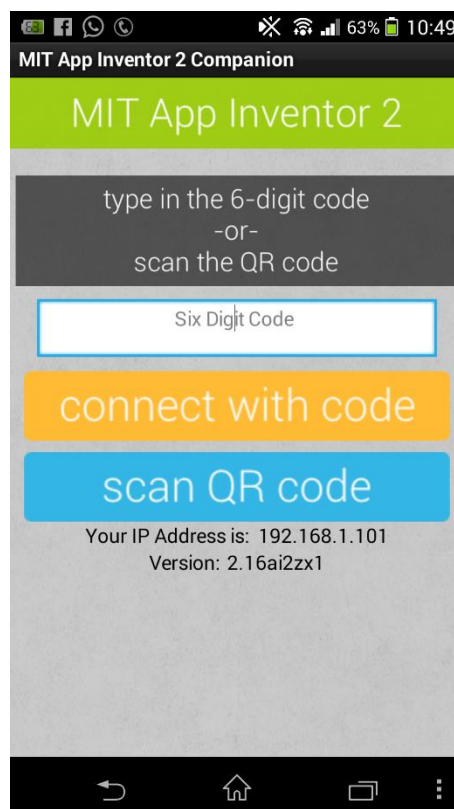
Step 3: 打開一個準備使用 WiFi 連線測試的 App Inventor 專案，此處以我們剛剛開發的「FirstApp」為例，點選執行功能列上的【Connect/AI Companion】選項，表示要以 WiFi 連線的方式來測試 App 程式，如圖所示。



Step 4: 此時電腦會出現一個「Connect to Companion」視窗，上面會有 QR code 及編碼，如圖所示。



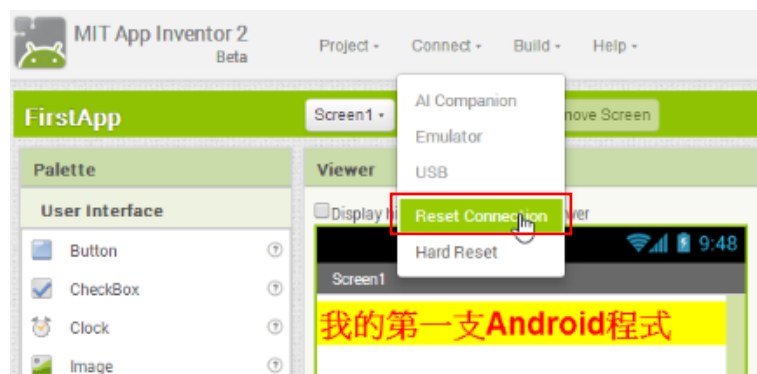
Step 5：接著執行 Android 裝置上的「MIT AI2 Companion」App 程式，使用其「scan QR code」按鈕來掃描上圖的 QR code，然後按下「connect with code」按鈕，就可以直接在 Android 裝置上測試執行程式。或者是在文字方塊內輸入上圖的 code，輸入完按下「connect with code」按鈕即可。



Step 6：最後，在 Android 裝置上會顯示「FirstApp」程式的測試執行結果，如圖所示。



當測試執行結束，我們可以點選執行功能列上的【Connect/Reset Connection】選項，來終止 WiFi 連線測試，如圖所示。

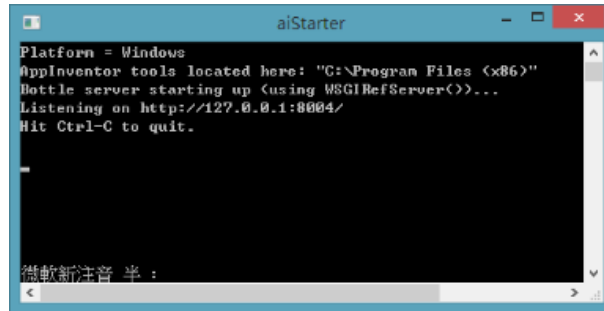


2. 模擬器方式

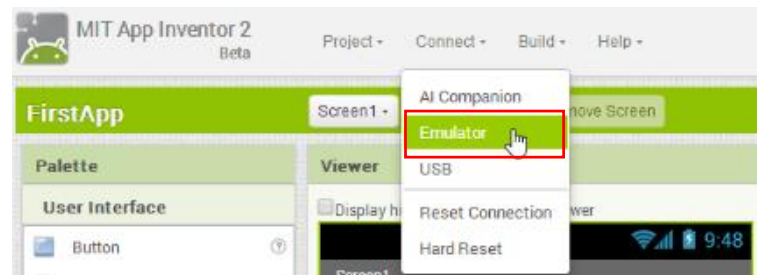
使用模擬器（Emulator）方式來測試執行開發好的程式，需要在電腦安裝 MIT App Inventor 2 軟體。

使用模擬器來測試執行程式的步驟如下：

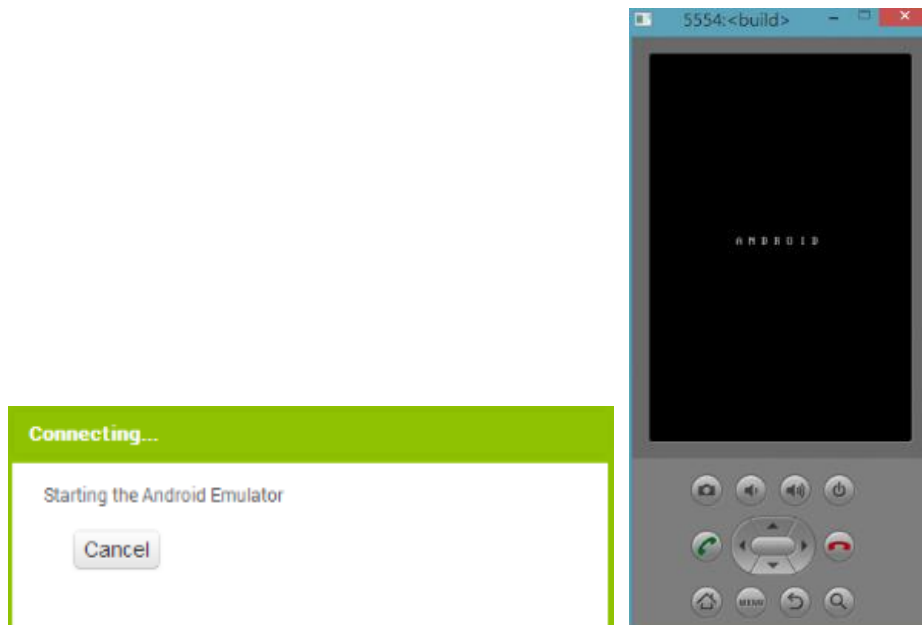
Step 1: 選取執行程式集裡的「aiStarter.exe」程式，執行後會出現一個命令視窗，當開始使用模擬器進行測試後，會在命令視窗出現相關資訊。



Step 2：打開一個準備使用模擬器測試的 App Inventor 專案，此處以我們剛剛開發的「FirstApp」為例，點選執行功能列上的【Connect/Emulator】選項，表示要以模擬器方式來測試 App 程式，如圖所示。



Step 3：出現正在嘗試連到模擬器的連接視窗（Connecting...），並出現模擬機的開機畫面如圖所示。

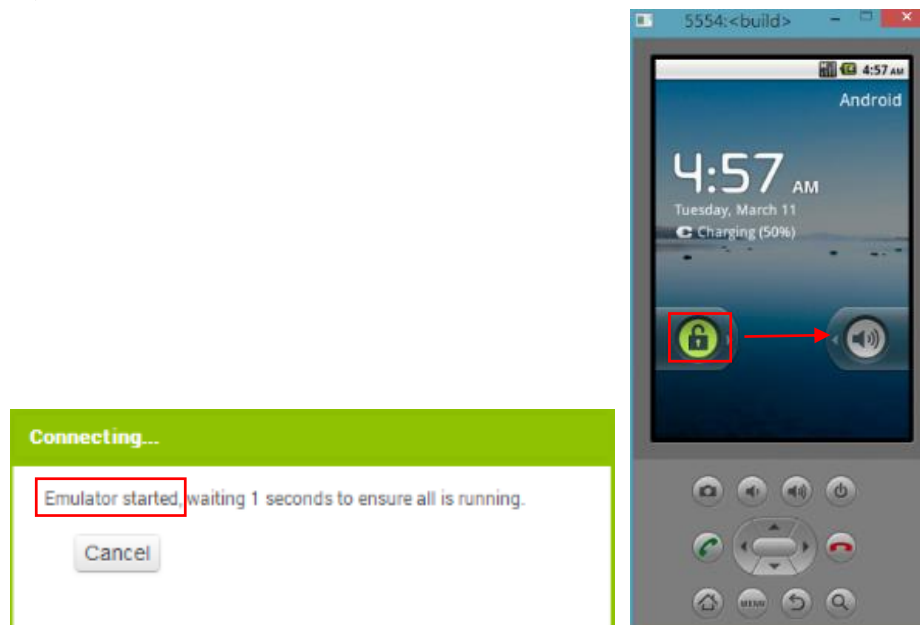


此時 aiStarter 命令視窗也會出現一些運作訊息，如圖所示。

```
aiStarter
Platform = Windows
Appinventor tools located here: "C:\Program Files (x86)"
Bottle server starting up (using WSGIServer())...
Listening on http://127.0.0.1:8004/
Hit Ctrl-C to quit.

127.0.0.1 - - [11/Mar/2014 12:57:22] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:22] "GET /start/ HTTP/1.1" 200 0
127.0.0.1 - - [11/Mar/2014 12:57:23] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:23] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:25] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:25] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:27] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:27] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:29] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:29] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:31] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:31] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:33] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:33] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:35] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:35] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [11/Mar/2014 12:57:37] "GET /echeck/ HTTP/1.1" 200 38
```

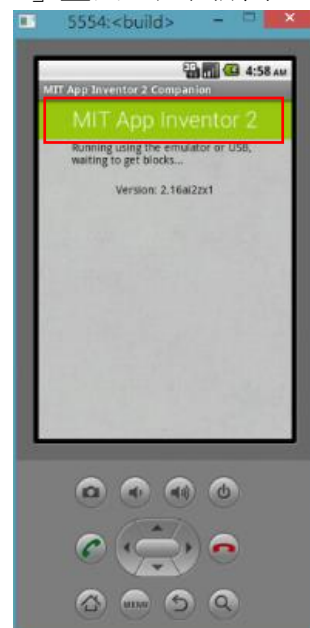
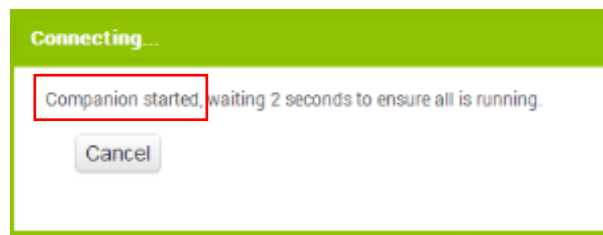
Step 4：當模擬器成功啟動，連接視窗會出現「Emulator started」文字，請將模擬機中的綠色鎖向右拖曳來解鎖，如果沒有進行解鎖，程式仍會繼續往下執行，如圖所示。



綠色鎖向右拖曳完成解鎖後，模擬機的畫面如圖所示。



Step 5：接著連接視窗會出現「Companion started」文字，表示模擬器已經啟動，此時模擬器的畫面會出現「MIT App Inventor 2」畫面，如圖所示。



Step 6：最後，在模擬器顯示出「FirstApp」程式的執行結果，如圖所示。



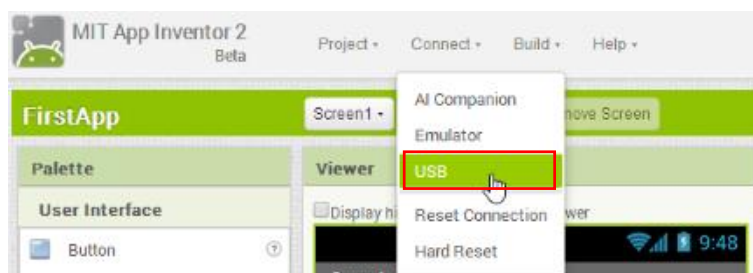
3.USB 連線

使用 USB 連線方式來測試執行開發好的程式，需要在電腦安裝 MIT App Inventor 2 軟體。

使用 USB 連線來測試執行程式的步驟如下：

Step 1: 選取執行程式集裡的「aiStarter.exe」程式，執行後會出現一個命令視窗，當開始使用模擬器進行測試後，會在命令視窗出現相關資訊。

Step 2：打開一個準備使用 USB 連線測試的 App Inventor 專案，此處以我們剛剛開發的「FirstApp」為例，點選執行功能列上的【Connect/USB】選項，表示要以 USB 連線方式來測試 App 程式，如圖所示。



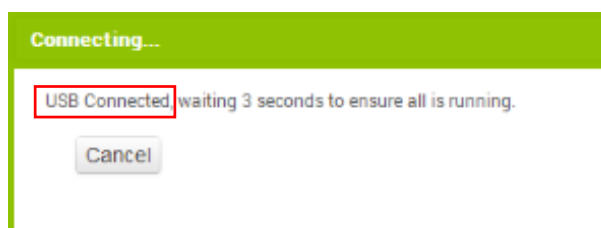
Step 3：接著連接視窗會出現嘗試透過 USB 連線的訊息（Connecting via USB Cable），表示要使用 USB 連線來測試執行程式，因此我們需要勾選 Android 裝置的「USB 偵錯」模式。



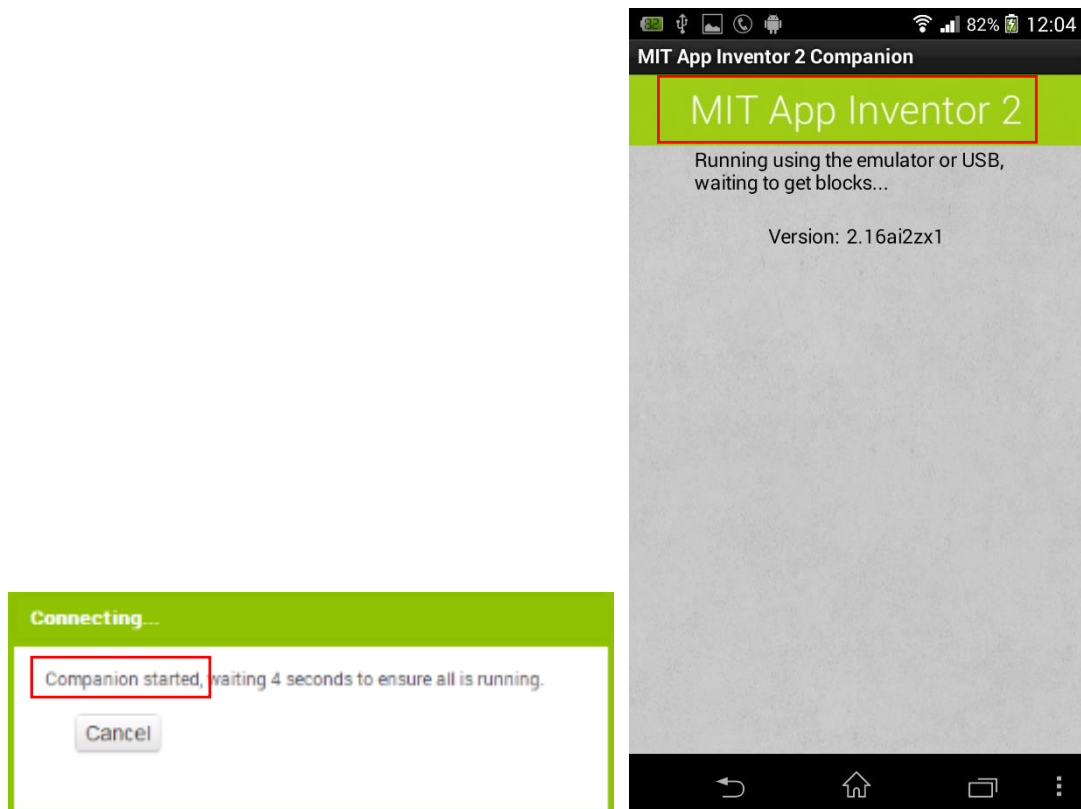
小提示：如何打開 Android 裝置的「USB 偵錯」模式？

如果 Android 裝置使用的 Android 版本是 3.2 版或更舊的版本，「USB 偵錯」模式的路徑是在【設定/應用程式/開發】，勾選 USB 偵錯，才能在使用 USB 連線時測試程式。如果 Android 版本是 4.0 版或更新的版本，「USB 偵錯」模式的路徑是在【設定/開發人員選項】。此處請特別注意，如果 Android 版本是 4.2 版或更新的版本，「開發人員選項」在預設狀況下是隱藏的，要顯示「開發人員選項」，需要在【設定/關於手機/軟體版本】選項上，連續輕觸數下，以顯示「開發人員選項」。

Step4：接著連接視窗會出現 USB 已經連線的訊息（USB Connected），如圖所示。



Step5：然後連接視窗會出現「Companion started」文字，同時 Android 裝置會顯示「MIT App Inventor 2」，表示已經透過 USB 連線的方式，準備好要執行 App 程式了。



Step 6：最後，在 Android 裝置上，顯示出「FirstApp」程式的執行結果，如圖所示。

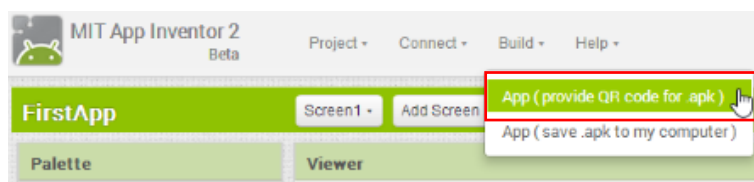


➤ 安裝程式到 Android 裝置

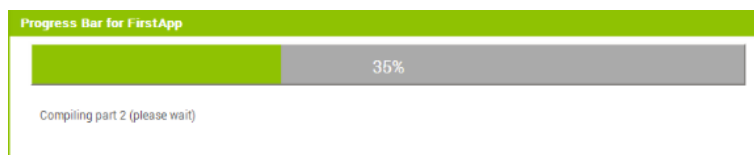
先前我們透過 WiFi 連線或是 USB 連線的方式，將 MIT App Inventor 2 的程式，放到 Android 裝置上去測試執行，此方式並不是真正安裝 App 程式到 Android 裝置上。

要將開發完成的 App 程式安裝到 Android 裝置上，需要將 App 程式建置成封裝檔（.apk）格式，然後安裝到 Android 裝置上，其步驟如下：

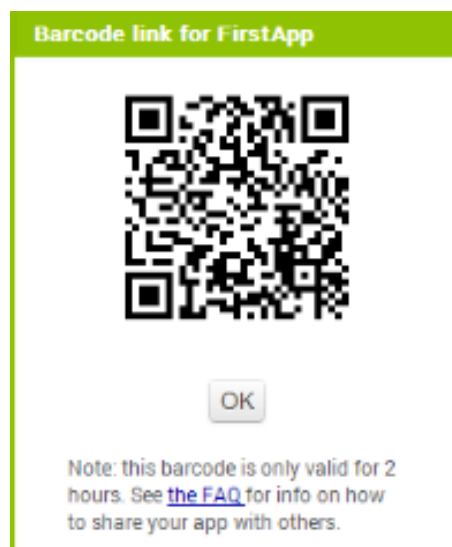
Step1：在 MIT App Inventor 2 網站，打開一個準備安裝到 Android 裝置的 App Inventor 專案，我們以剛剛開發的「FirstApp」為例，點選執行功能列上的【Build/App (provide QR code for .apk)】選項，可以建置.apk 安裝檔，如圖所示。



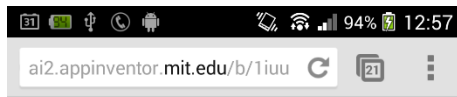
Step2：出現進度視窗，顯示出目前建置的進度。



Step3：當進度達到 100%後，會出現這個 App 程式專屬的 QR code。



Step4：接下來我們可以使用 Android 裝置的 QR code 掃描程式（例如：Play 商店中的 QR Droid 程式，可自 Play 商店中免費下載使用），掃描上圖的 QR code，掃描後會連結到一個網頁，詢問是否要保留「FirstApp.apk」，此處我們選擇「確定」；接著出現是否要安裝應用程式畫面，此處我們選擇「安裝」。

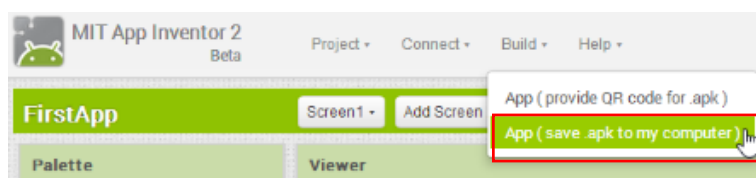


Step5：應用程式安裝完成後，會在 Android 裝置上出現「FirstApp」圖示，點選圖示即會執行此應用程式，其執行結果如下圖，成功完成安裝 App 應用程式到 Android 裝置。

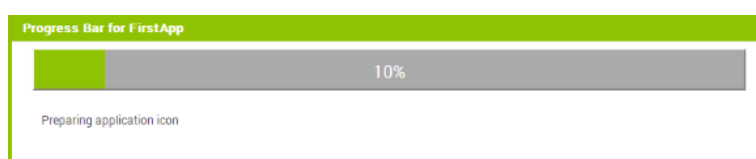


另一種方式是將封裝檔 (.apk) 下載到電腦後，將檔案複製到 Android 裝置，再進行安裝的動作，其步驟如下：

Step1：點選執行功能列上的【Build/App (save .apk to my computer)】選項，如圖所示。



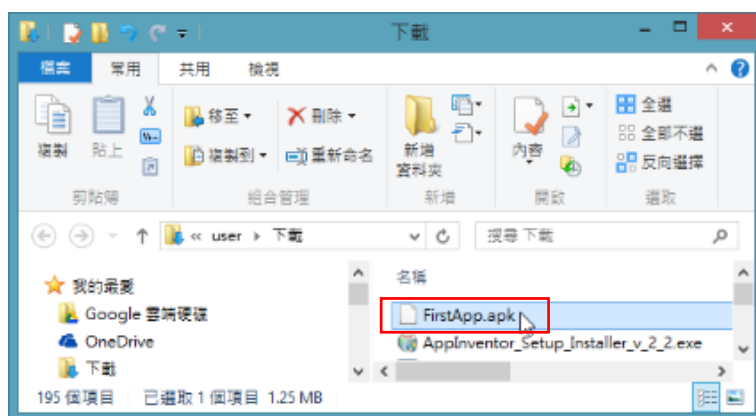
Step2：出現進度視窗，顯示出目前建置的進度。



Step3：當進度達到 100%後，會下載「FirstApp.apk」檔案到下載資料夾內。



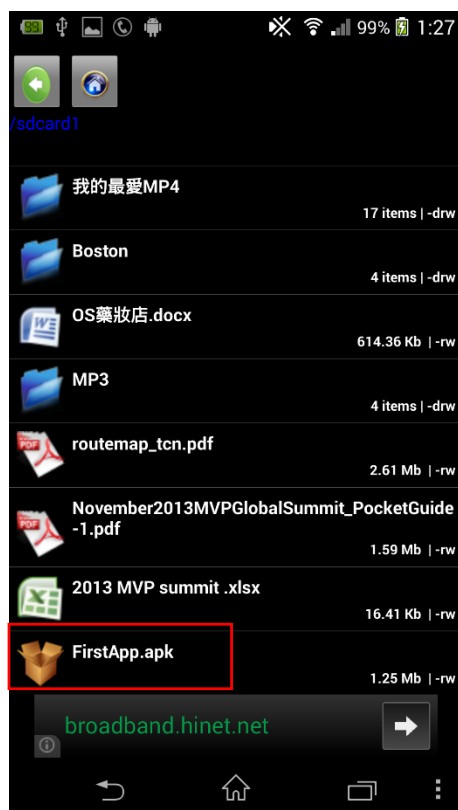
Step4：檢視下載資料夾，果然出現「FirstApp.apk」封裝檔，我們可以將此封裝檔複製到 Android 裝置，然後再以可以安裝 APK 類型檔案的應用程式，來安裝「FirstApp.apk」。



Step5：我們可以到 Play 商店中，搜尋可以安裝 APK 封裝檔的應用程式，例如：輸入「apk installer」來搜尋，選擇其中一個應用程式下載安裝後，就可以進一步安裝「FirstApp.apk」封裝檔了。在 Play 商店中搜尋「apk installer」的結果如圖所示，我們選擇安裝第一個應用程式。



下圖為打開 Apk Installer 應用程式後的畫面，在 Android 裝置中，找到「FirstApp.apk」檔案，然後點擊其圖示來安裝應用程式。



Step6：應用程式安裝完成後，會在 Android 裝置上出現「FirstApp」圖示，點選圖示即會執行此應用程式，其執行結果如下圖，成功完成安裝 App 應用程式

到 Android 裝置。



使用 MIT App Inventor 2 網站提供的雲端服務，以拼圖模式來組程式，讓我們可以免費使用其提供的強大功能，進而設計 Android 裝置上的 App 應用程式，完成作品後，還可以上傳到 Google Play 商店，提供給廣大的使用者下載安裝，讓我們動手來開發 Android 裝置上的 App 應用程式吧！

➤ 拼圖塊程式範例 1：百貨公司打折 App

一、程式設計目標

本範例希望設計一個多段式百貨公司週年慶打折程式。設計一個文字方塊，讓使用者輸入在百貨公司購買的金額，然後按下「計算」按鈕，App 經過計算後，會告訴使用者打折後的價格為何。

以下為此 App 對購買金額的判別規則：消費 3000 元以內，打 9 折；消費 3000～6000 元，前 3000 元打 9 折，3000～6000 元打 8 折；消費 6000 元以上，前 3000 元打 9 折，3000～6000 元打 8 折，超過 6000 元打 7 折。

例如：輸入 2800，打折後會得到 2520 元。



如果輸入 4800，打折後會得到 4140 元。



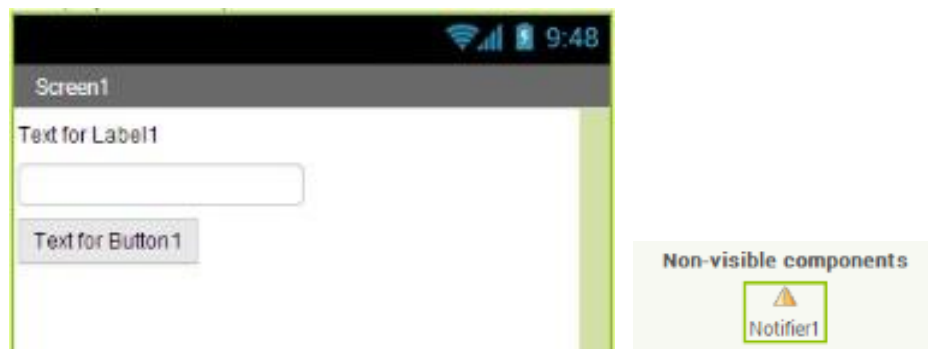
如果輸入 7000，打折後會得到 5800 元。



二、螢幕配置圖例

在 Screen 元件上，布置 1 個 Label 元件、1 個 TextBox 元件、1 個 Button 元件

及 1 個 Notifier 元件，如圖所示。



三、屬性值設定

屬性值設定表格：

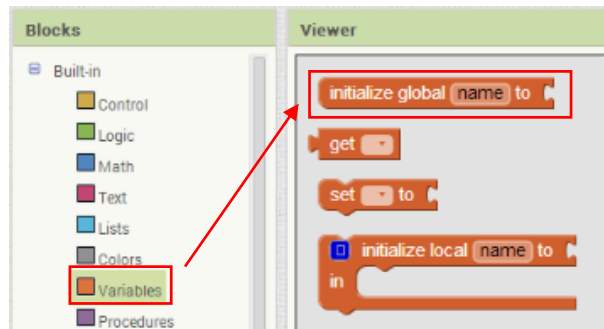
元件	屬性	屬性值
Screen1	Title	超級百貨公司打折 App
Label1	Text	請輸入消費金額：
TextBox1	Hint	清空
	NumbersOnly	勾選
Button1	Text	計算

屬性值設定後的畫面如下：



四、拼圖 Blocks 編輯

Step1：點選 Blocks 頁面的「Blocks 面板」內之 Variables 元件，接著再點選 Blocks 頁面的「Viewer 面板」，選擇其中的「initialize global name to」拼圖塊，設定全域變數的初值。



此時我們設定 2 個全域變數，一個名稱為「Spend_M」代表消費者的消費金額，另一個名稱為「Money」代表打完折後的總價，並令這兩個變數的初始值為「0」，如圖所示。



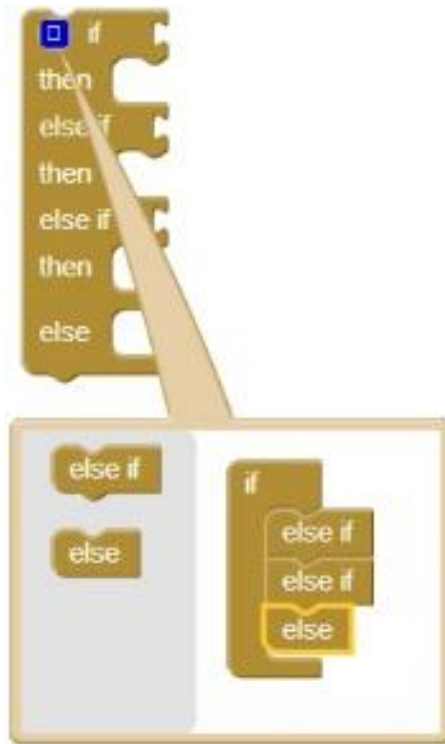
Step2：點選 Blocks 頁面的「Blocks 面板」內之 Button1 元件，接著再點選 Blocks 頁面的「Viewer 面板」，選擇其中的「when Button1.Click do」拼圖塊，設定 Button1 元件被按下時的事件動作。



Step3：將「Spend_M」全域變數的值設定為 TextBox1 的 Text 值。



Step4：使用「If...Then...」拼圖塊，按下藍色方塊後，從中拖曳出 2 個「Elseif」及 1 個「Else」拼圖塊，如圖所示。



Step5：依據百貨公司的打折規則，先處理消費在 3000 元以內的部分。此部分的判斷，我們使用 **and** 運算來處理，當消費金額確實落在 0~3000 元之間時，才去把消費金額打 9 折（乘以 0.9），運算後把金額指定給「Money」變數。最後使用 **Notifier** 元件顯示出「Money」變數的值，拼圖塊如下所示。



Step6：接下來處理消費落在 3001~6000 元之間的狀況。依據規則，前 3000 元打 9 折（也就是 2700 元），落在 3000~6000 元的部分打 8 折（乘以 0.8），此部分的拼圖塊如下所示。



Step7：最後來處理消費超過 6000 元的狀況。依據規則，前 3000 元打 9 折（也就是 2700 元），落在 3000~6000 元的部分打 8 折（也就是 2400 元），兩

部分加起來 5100 元，超過 6000 元的部分打 7 折（乘以 0.7），此部分的拼圖塊如下所示。



Step8：最後是處理特殊狀況的部分，當消費者輸入負值或其他非正確的文字，此時會出現警告訊息，此部分的拼圖塊如下所示。



Step9：將 Step3~Step8 的所有拼圖塊，都置於「when Button1.Click do」拼圖塊之中，完成此 App 的設計。

➤ 拼圖塊程式範例 2：高鐵票價查詢 App

一、程式設計目標

本範例希望設計一個高鐵票價查詢程式，讓使用者輸入起始站編號和目的站編號，然後按下「計算」按鈕，最後告訴使用者應該要付多少票價。以下為台灣高鐵的票價表：

車站	台北	板橋	桃園
台北	--	45	175
板橋	45	--	140
桃園	175	140	--

以下用幾個輸入做舉例：

1. 若使用者輸入起始站為 3，目的站為 1，程式會依據台灣高鐵的票價表計算出票價為「175 元」。



2. 若使用者輸入起始站為 2，目的站為 2。由於起始站和目的站是同一個，所以會出現錯誤訊息。

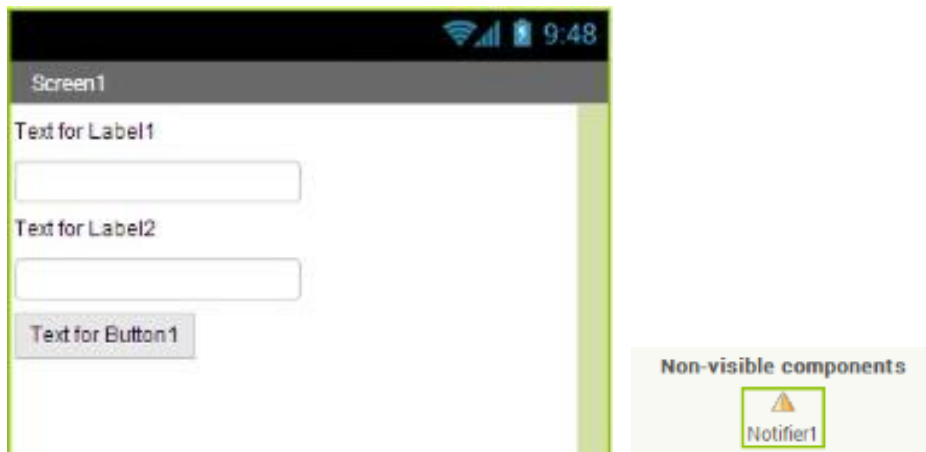


3. 若使用者輸入起始站為 5，目的站為 1，由於起始站超過整數 1~3 的範圍，所以會出現錯誤訊息。



二、螢幕配置圖例

在 Screen 元件上，布置 2 個 Label 元件、2 個 TextBox 元件、1 個 Button 元件及 1 個 Notifier 元件，如圖所示。



三、屬性值設定

屬性值設定表格：

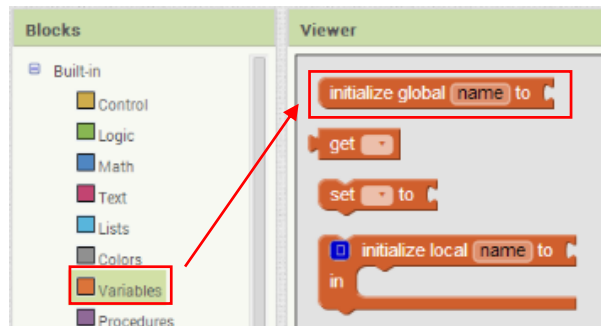
元件	屬性	屬性值
Screen1	Title	高鐵票價查詢 App
Label1	Text	請輸入起始站位置[1:台北 2:板橋 3:桃園]
TextBox1	Hint	清空
	NumbersOnly	勾選
Label2	Text	請輸入目的站位置[1:台北 2:板橋 3:桃園]
TextBox2	Hint	清空
	NumbersOnly	勾選
Button1	Text	計算

屬性值設定後的畫面如下：



四、拼圖 Blocks 編輯

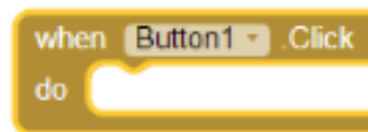
Step1：點選 Blocks 頁面的「Blocks 面板」內之 Variables 元件，接著再點選 Blocks 頁面的「Viewer 面板」，選擇其中的「initialize global name to」拼圖塊，設定全域變數的初值。



此時我們設定 2 個全域變數，一個名稱為「Start」代表起始站，另一個名稱為「End」代表目的站，並令這兩個變數的初始值為「0」，如圖所示。



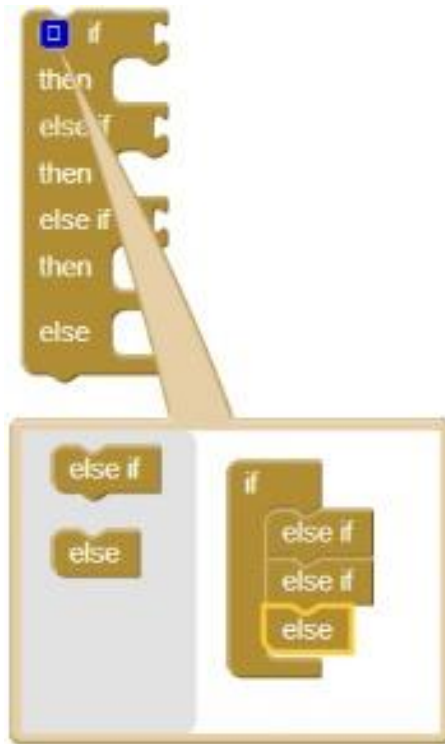
Step2：點選 Blocks 頁面的「Blocks 面板」內之 Button1 元件，接著再點選 Blocks 頁面的「Viewer 面板」，選擇其中的「when Button1.Click do」拼圖塊，設定 Button1 元件被按下時的事件動作。



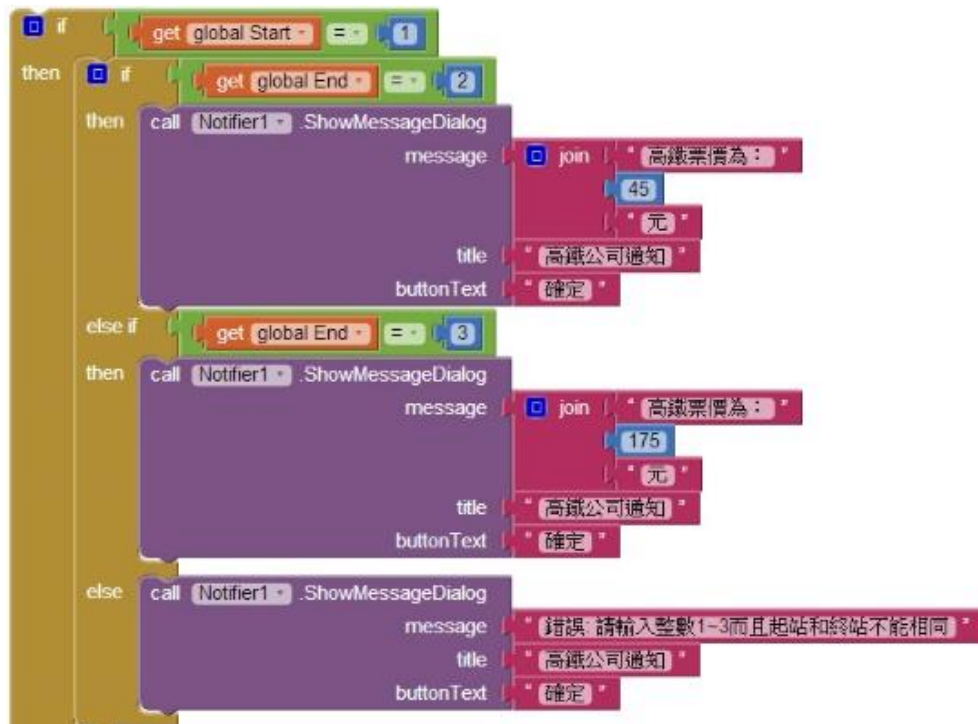
Step3：將「Start」全域變數的值設定為 TextBox1 的 Text 值，儲存起始站的資料，將「End」全域變數的值設定為 TextBox2 的 Text 值，儲存目的站的資料。



Step4：使用「If...Then...」拼圖塊，按下藍色方塊後，從中拖曳出 2 個「Elseif」及 1 個「Else」拼圖塊，如圖所示。

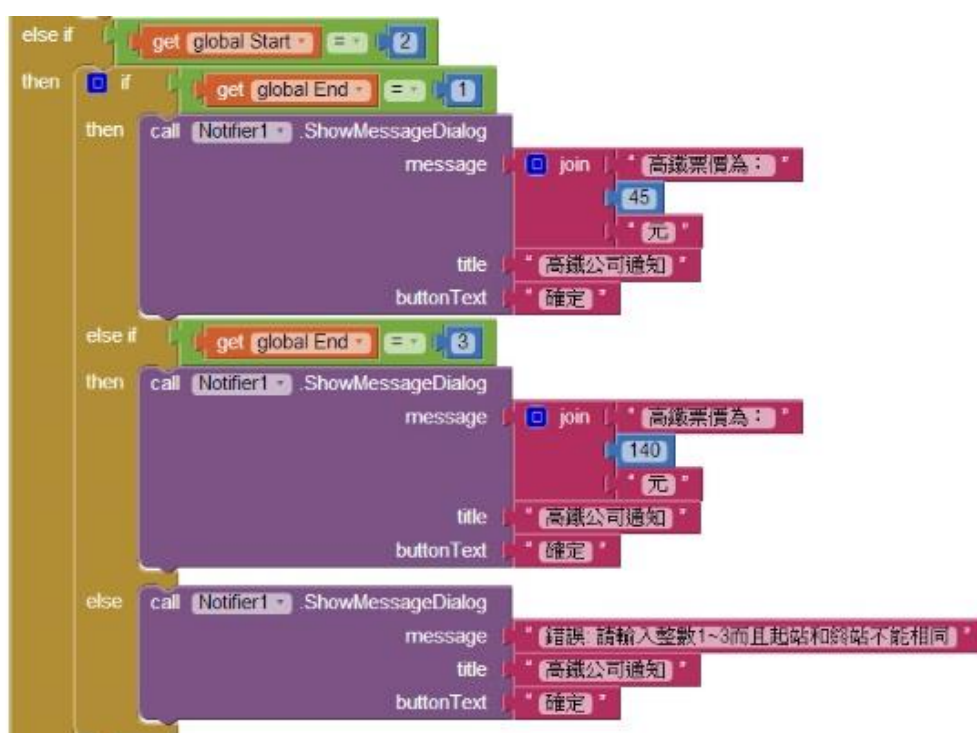


Step5：此處我們使用巢狀判斷結構，外層「If...Then...ElseIf...Else...」判斷起始站的站別，內層「If...Then...ElseIf...Else...」判斷目的站的站別，依據使用者輸入的起始站及目的站號碼來判斷車價，最後使用 Notifier 元件顯示出該付的費用。我們先處理起始站為「1」，目的站為「2」、「3」或「其他」的狀況，拼圖塊如下所示。

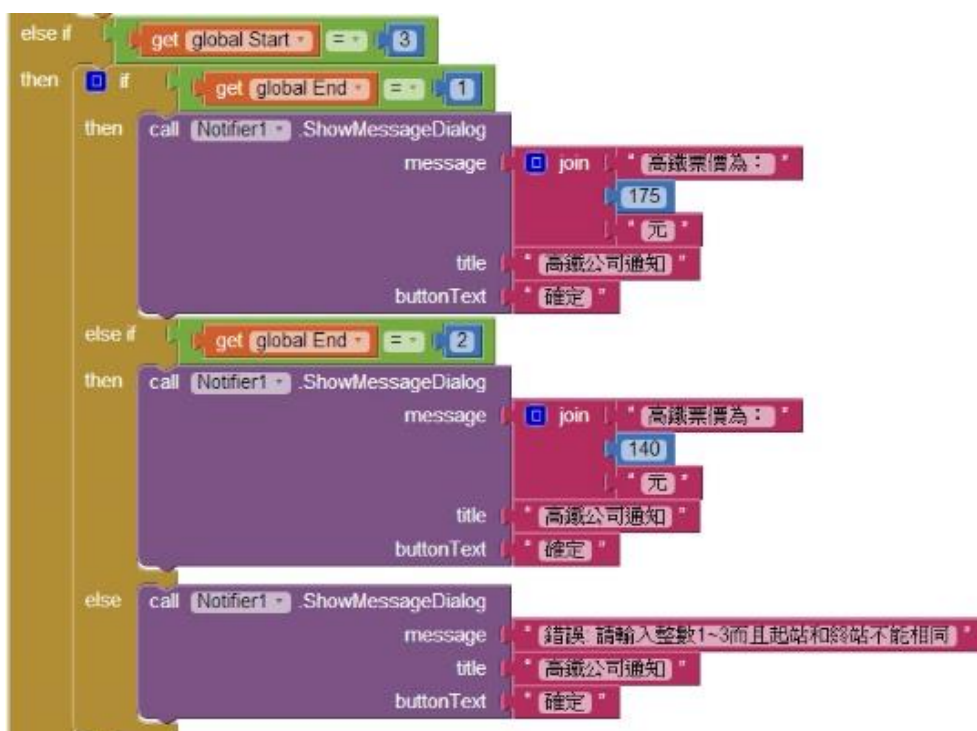


Step6：接著處理起始站為「2」，目的站為「1」、「3」或「其他」的狀況，拼

圖塊如下所示。



Step7：接著處理起始站為「3」，目的站為「1」、「2」或「其他」的狀況，拼圖塊如下所示。



Step8：最後是處理特殊狀況的部分，當消費者輸入其他的內容時，會出現提示文字，此部分的拼圖塊如下所示。



Step9：將 Step3~Step8 的所有拼圖塊，都置於「when Button1.Click do」拼圖塊之中，完成此 App 的設計。