

修平科技大學

資訊網路技術系實務專題

健康管理計步器 App

指導老師：蔡篤校老師

組員：BN100094 蔡庭暉

BN100095 黃彥智

BN100105 蔡似易

中華民國 104 年 6 月 17 日

修平科技大學

資訊網路技術系實務專題

健康管理計步器 App

組員：BN100094 蔡庭暉

BN100095 黃彥智

BN100105 蔡似易

指導老師：_____老師

評審老師：_____老師

_____老師

_____老師

_____老師

中華民國 104 年 6 月 17 日

致謝

在此感謝蔡篤校老師在一年半的專題課程中，給予我們很多的知識與建議，我們在專題裡常常遇到很多專業的問題，老師也能耐心的回應與指導，老師平時除了要忙學校的事務，忙課程的問題和一些公事總是不厭其煩的指導，還是盡可能協助我們解決問題並也付出了許多時間與耐心與我們討論。

這一年半我們利用密集的討論時間，我們從什麼都不了解，到藉由此專題將我們起伏不定的玩樂之心，凝聚了共心力，即便僅剩茶餘飯後的零碎時間，我們也毫無浪費，除了在討論之餘，回家也緊湊練習Android APP Inventor最後還是謝謝一起完成專題的組員，在經過這將近一年半的時間努力與合作，雖然中間遇到程式轉換上的瓶頸、程式運用上的挫折、想法與意見上爭執，但大家仍願意凝聚一起，同心協力共同完成專題，經過許多努力的付出，最後終於有成品可以呈現給各位評審委員，相信製作專題的過程及團隊努力的結果，我們最終必定是滿載收獲

目錄

目錄.....	- 2 -
摘要.....	- 4 -
第一章、緒論	- 5 -
(一)、研究背景.....	- 5 -
(二)、研究動機.....	- 6 -
(三)、研究目的.....	- 7 -
(四)、系統流程圖	- 8 -
第二章、系統開發平台介紹.....	- 9 -
(一)、Android 介紹.....	- 9 -
(二)、Inventor 2 系統介紹.....	- 9 -
1. 安裝 App Inventor 2 開發工具	- 10 -
2. App Inventor 2 專題製作、管理.....	- 14 -
3. App Inventor 2 介面介紹.....	- 18 -
4. App Inventor 2 Screen 元件	- 20 -
5. App Inventor 2 Lable 元件.....	- 23 -
6. 拼塊編輯頁面.....	- 24 -
7. 建立模擬器	- 25 -
8. 在實機上執行.....	- 26 -
第三章、系統操作介面	- 27 -
(一)、開啟.....	- 27 -
(二)、第一次使用必須先設定基本資料.....	- 28 -
(三)、主畫面顯示使用者輸入的資訊.....	- 29 -
(四)、【健康管理】	- 30 -
(五)、【減重計畫】	- 31 -
(六)、【計步器】前置設定.....	- 32 -

(七)、【計步器】開始使用	- 33 -
(八)、【查看歷史資料】	- 35 -
第四章、完成目標與結論心得	- 27 -
(一)、完成目標.....	- 37 -
(二)、未來展望.....	- 37 -
第六章、參考文獻	- 38 -
附錄.....	- 39 -
(一)、頁面配置.....	- 39 -
(二)、系統程式.....	- 43 -

摘要

健康管理計步器是以使用者走路所產生的振動來計算步數，所以必須精準的偵測震動，而且每個使用者的震動程度不盡相同，本專題設計 10 種不同靈敏度，如果預設的靈敏度不準確，使用者可自行調整。

本專題提供 BMI、BMR、標準體重、減肥、累積同一天走路步數功能，做為使用者判斷參考。歷史資料功能則會紀錄六十天步行資料，提供自我健康管理的參考。

第一章、緒論

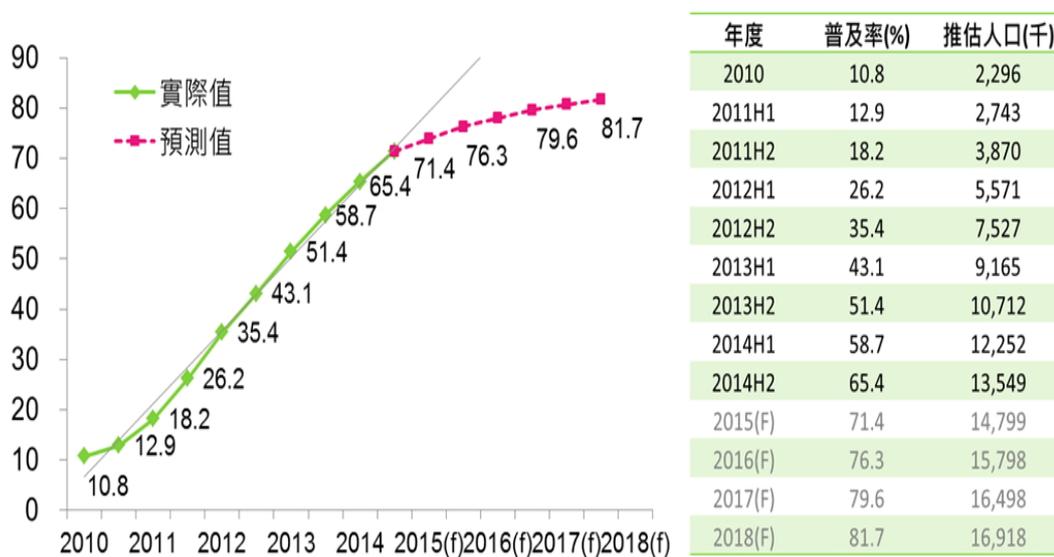
(一)、研究背景

在市場上的需求下，智慧型手機的品牌機型也就越來越多樣化，功能也就更強，在這樣的市場環境下，價格當然也會隨著各家廠商的競爭，

出現吸引消費者購買意願的價格了。

根據資策會在 2013 年的調查研究，2013 年的智慧型手機的普及率已經達到 43.1%，資策會更預估 2015 年智慧型手機的普及率將達到 67.2%，這是非常高的比率，所以智慧型手機的使用者只會越來越多，而智慧型手機的應用程式為響應越來越多的使用者，將會設計更好、

更方便及更有趣的手機應用程式。



(圖 1.1 智慧型手機普及率及成長趨勢(來源：財團法人資訊工業策進會))

(二)、研究動機

根據臨床研究，每日步行數達八千者，其罹患高血壓的機率降低80%。配合此一需求，應用程式是要設計成可以累計同一天行走的步數，

讓使用者查看數據就能知道當天是否達到預定目標。

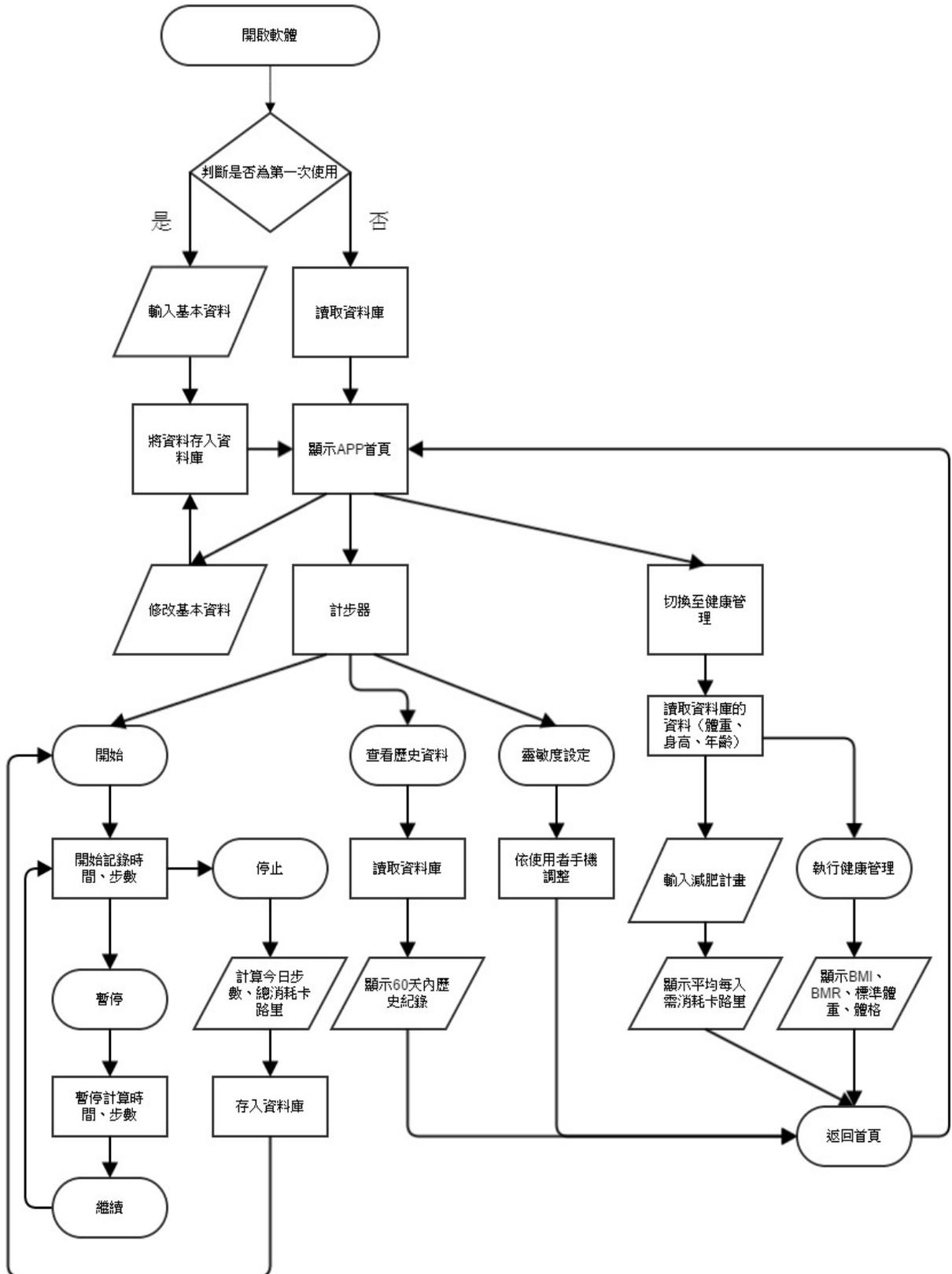
隨著現今社會的普及，人手一支智慧型手機的趨勢已經是相當平凡了，人們追求智慧型手機帶來的便利，現代人最重視的是莫過於健康的身體，而已步行增進健康是大多數健康專家推薦的方式，因為不須學習，也不太受場地限制，即使天氣不佳也可以在室內實施，若能於大自然中行走，一方面消耗熱量，一方面欣賞天然美景，對健康注

意更大。

(三)、研究目的

坊間有販售各式計步器，如果手機能設計計步器應用程式，不但可以節省購買計步器的費用，而且可以隨時隨地使用，不必擔心忘記攜帶計步器，一舉兩得。智慧型手機內建加速度感測器元件，步行時人體會自然產生震動，加速度感測器中 X、Y、Z 軸的加速度值會改變，利用此變化值即可偵測使用者的行走步數。

(四)、系統流程圖



(圖 1.2 系統流程圖)

第二章、系統開發平台介紹

(一)、Android 介紹

Android，中文俗稱安卓，是一個以 Linux 為基礎的開放原始碼行動裝置作業系統，主要用於智慧型手機和平板電腦，由 Google 成立的 Open Handset Alliance (OHA，開放手機聯盟) 持續領導與開發中。

(二)、Inventor 2 系統介紹

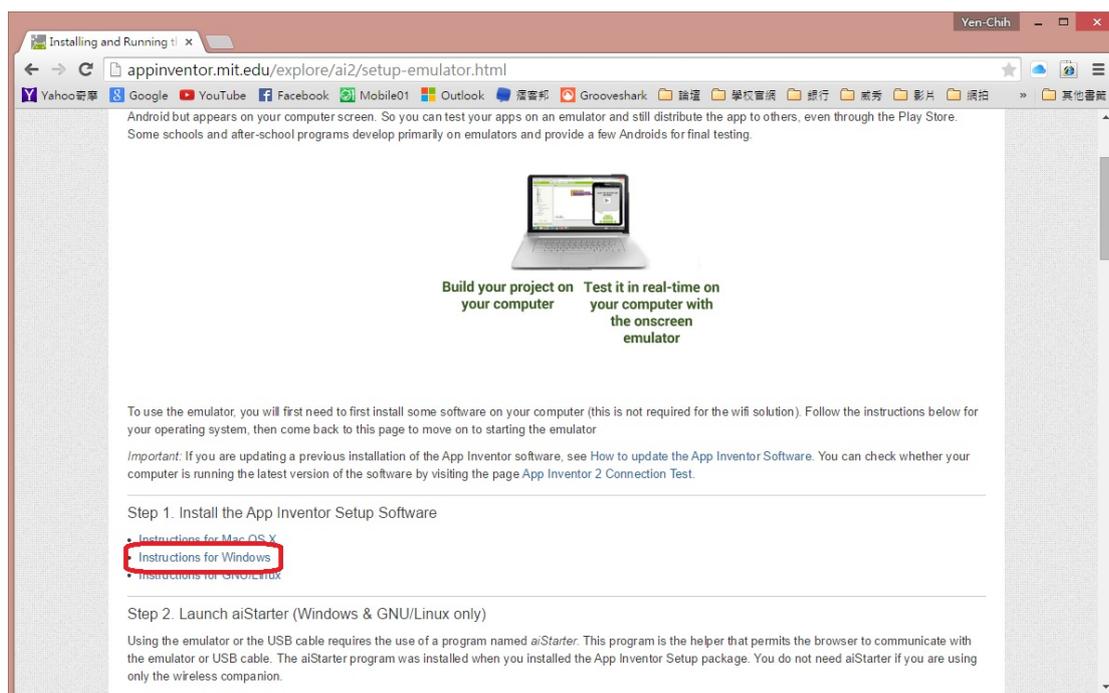
本系統我們所使用的開發工具是 App Inventor 2，加速度感測器多用於遊戲中控制遊戲角色的移動，也可用來測量行動裝置的震動狀況，App Inventor 2 貼心的提供了 Shaking 方法來偵測行動裝置的震動，但 Shaking 方法無法判別震動的度，不能用於計步器。健康計步器是以使用者走路產生的震動來計算步數，所以必須很精準的偵測震動，而且每個使用者的震動程度不盡相同，最好還是提供使用者能手動微調的功能。

1. 安裝 App Inventor 2 開發工具

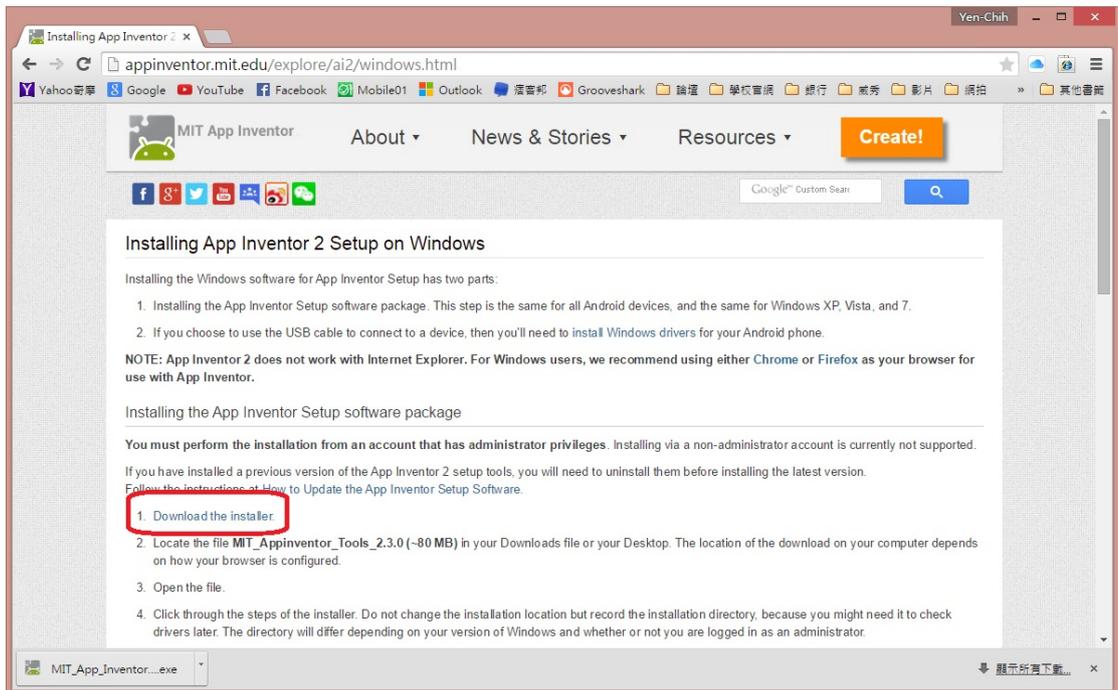
(1) 開啟網址並且下載

「[http://appinventor.mit.edu/explore/ai2/setup-emulator.ht](http://appinventor.mit.edu/explore/ai2/setup-emulator.html)

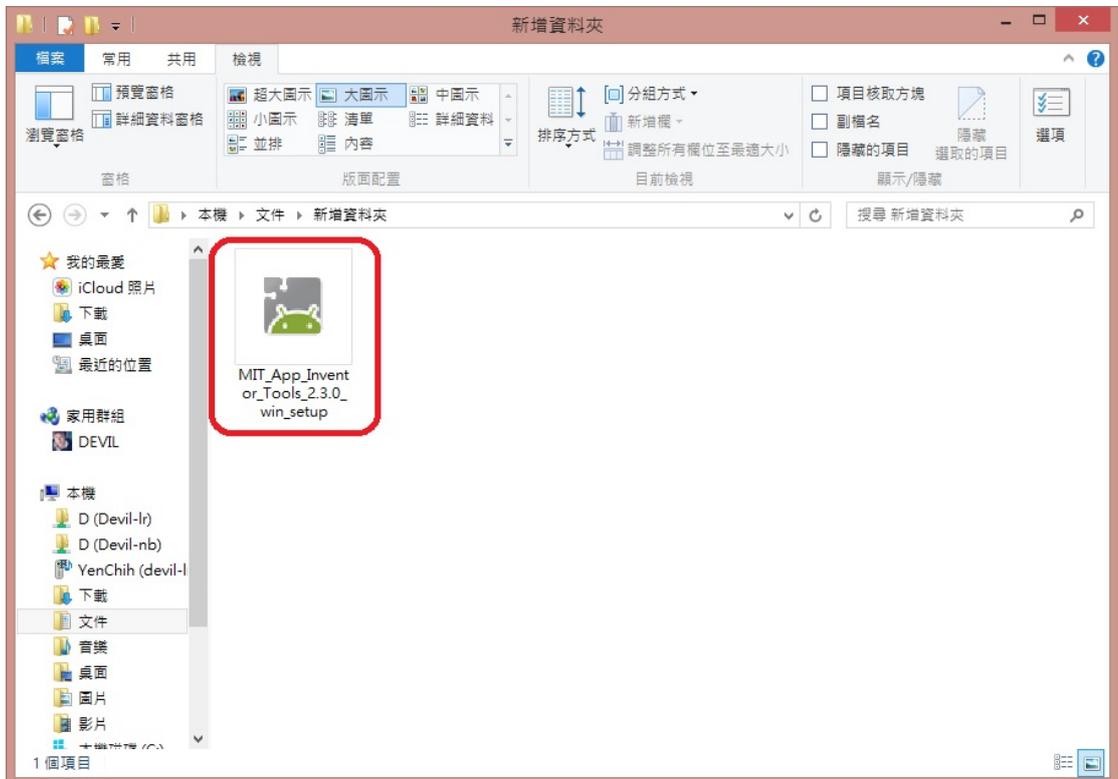
[ml](#)」點選下方【Instructions for Windows】。



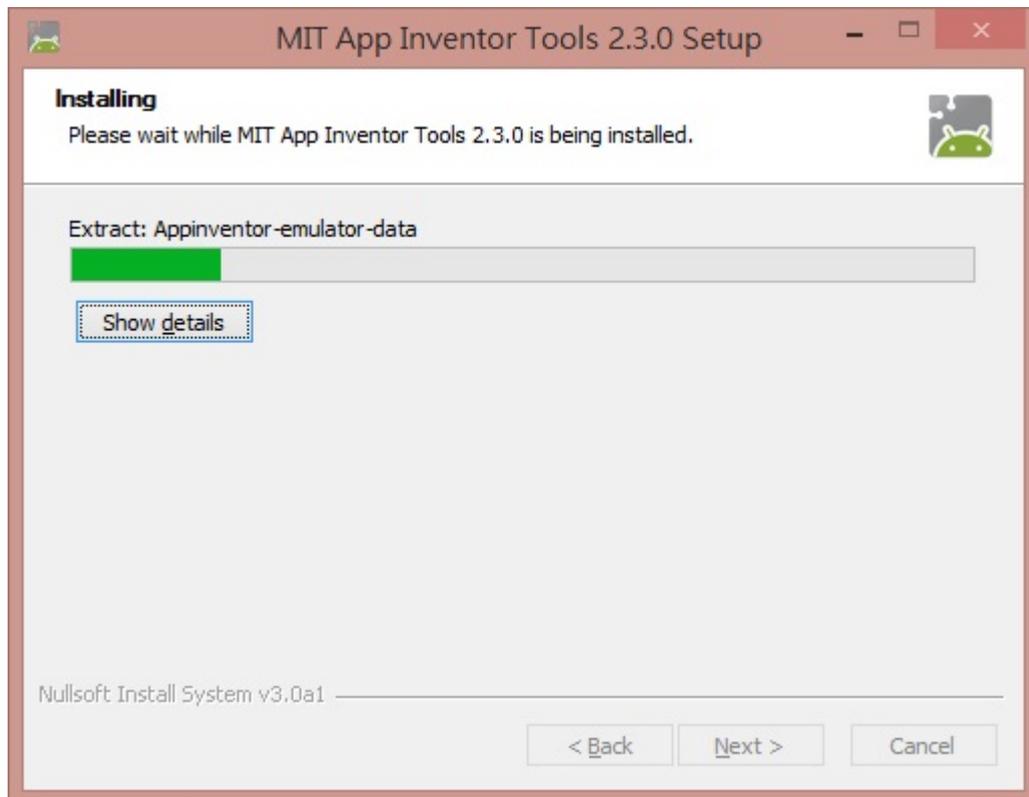
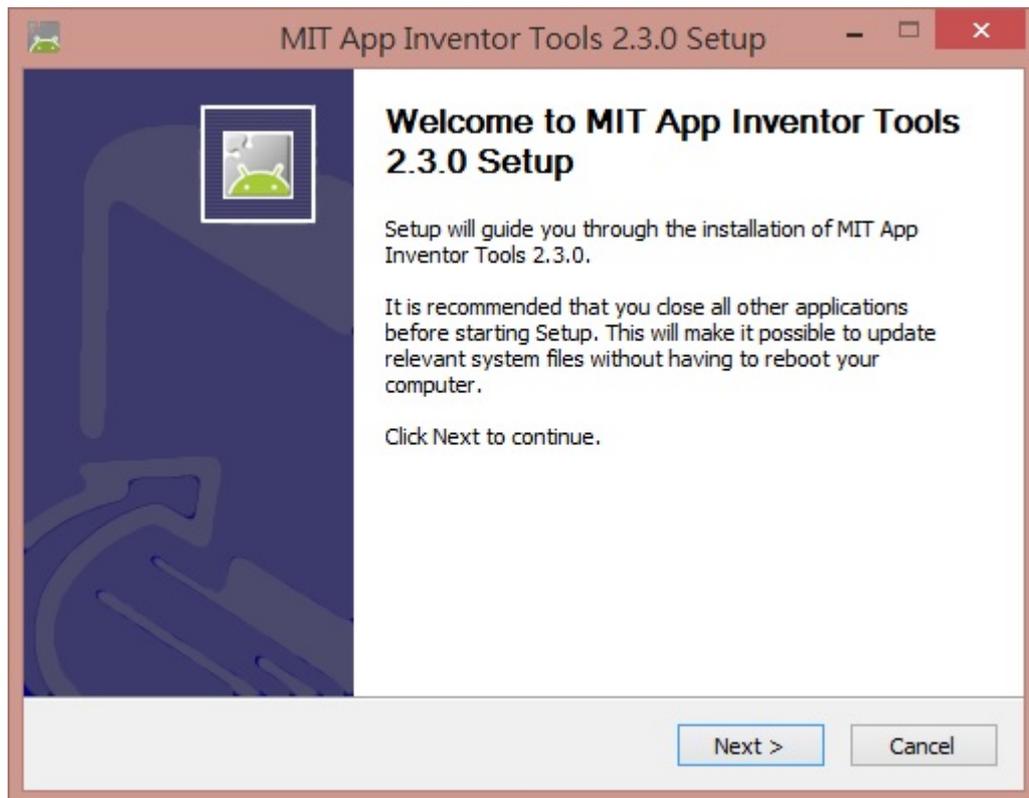
(2)於網頁中點選【Download the installer】下載檔案。

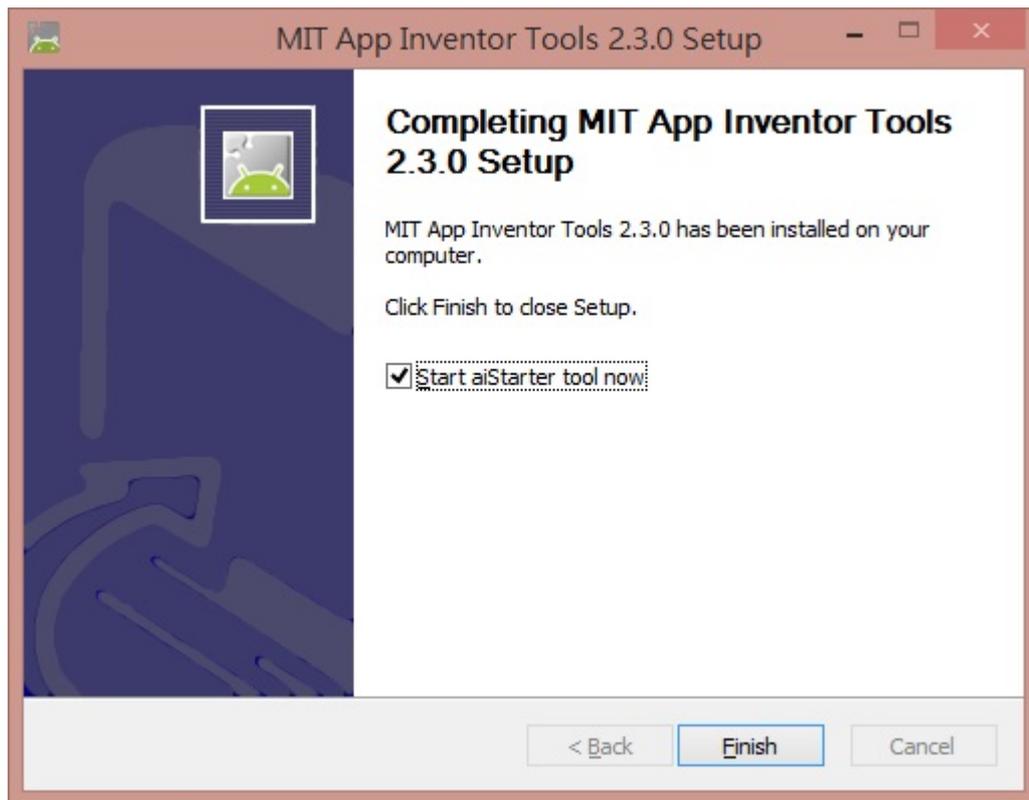


(3)下載完後開始安裝。



★安裝過程無須任何設定。



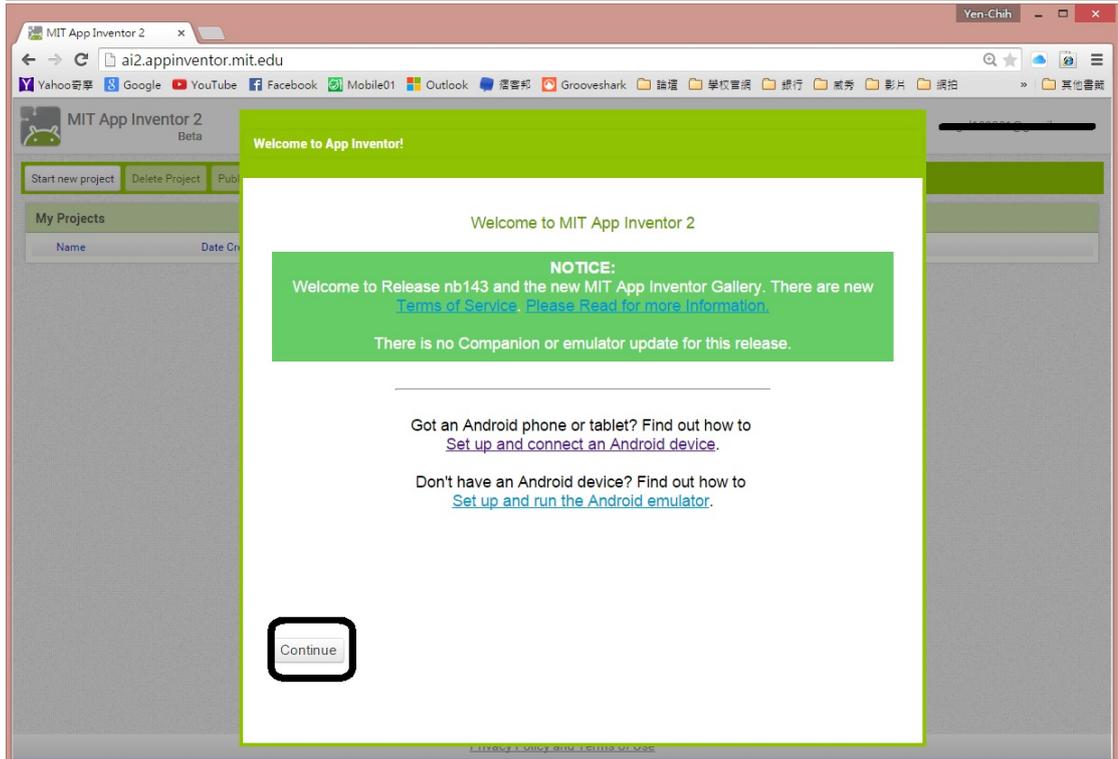
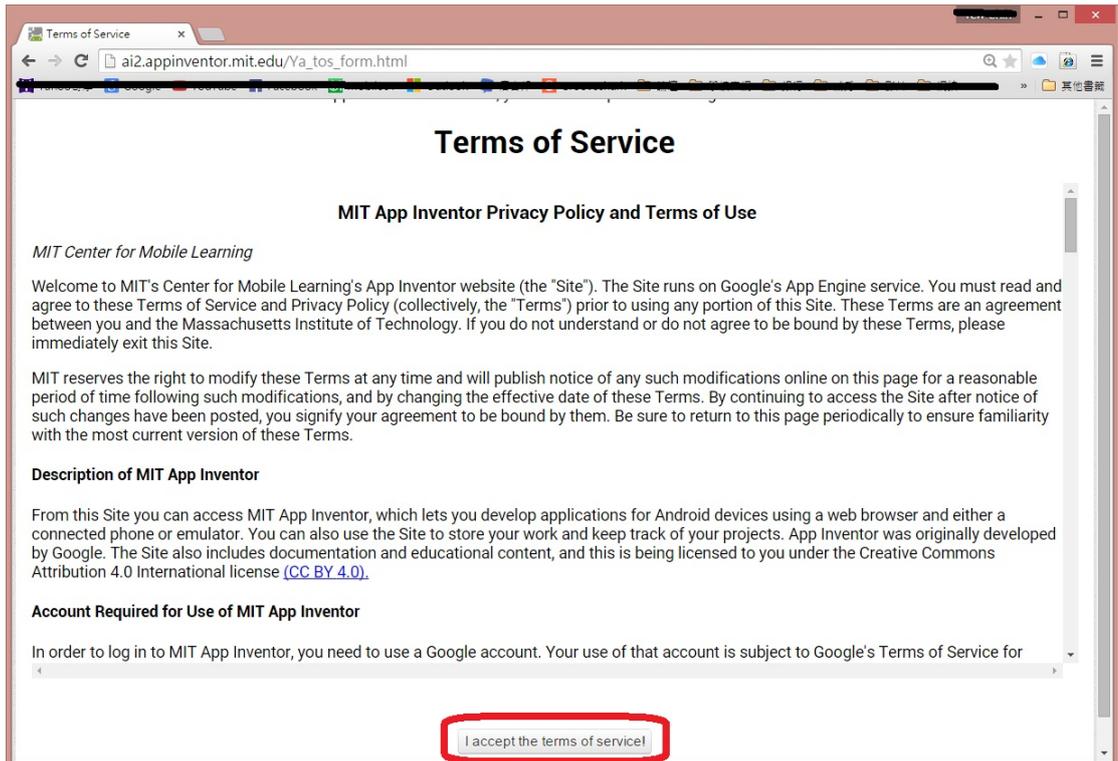


(4) 以上安裝完之後就可以開始 APP 應用程式設計了！

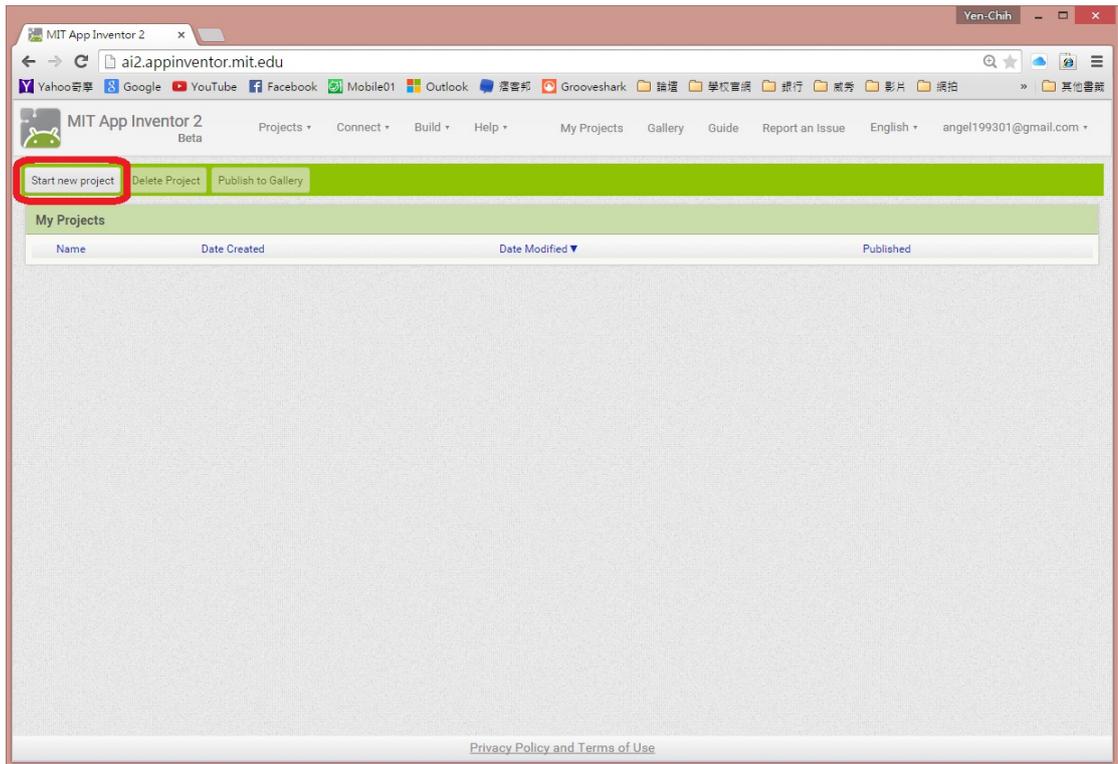
2. App Inventor 2 專題製作、管理

(1) 開啟網址「<http://ai2.appinventor.mit.edu/>」，登入帳號。

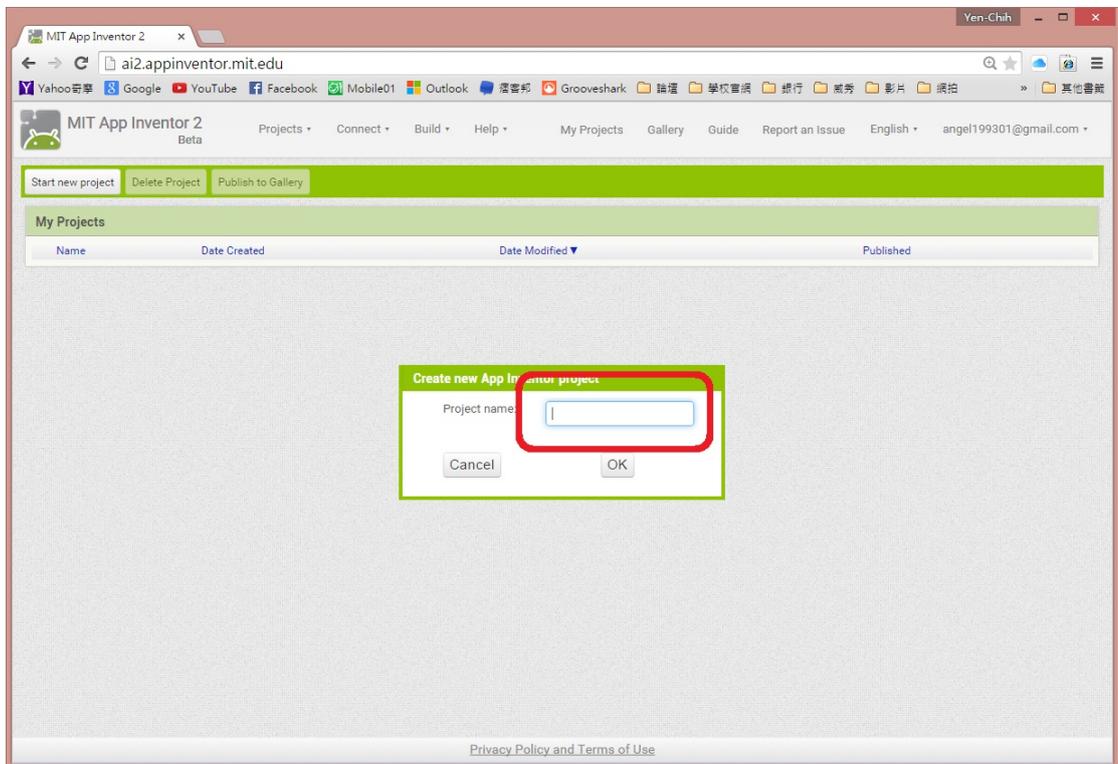
The image consists of two screenshots of a web browser. The top screenshot shows the Google login page. The browser's address bar displays the URL https://accounts.google.com/ServiceLogin?sacu=1&continue=https%3A%2F%2Fappengine.google.com%2F_ah%2Fconlogin. The page features the Google logo and the text "只要一個帳戶，所有 Google 服務暢行無阻。" and "使用您的 Google 帳戶登入". Below this is a login form with fields for "電子郵件" (Email) and "密碼" (Password), a blue "登入" (Sign In) button, and checkboxes for "保持登入狀態" (Keep me signed in) and "需要協助嗎?" (Need help?). A "建立帳戶" (Create account) link is also visible. The bottom screenshot shows the Google Accounts permission screen. The browser's address bar displays the URL https://appengine.google.com/_ah/loginform?state=AJKIYcGM5WtPsRtIIOrzBpAgEtUH-LN6Kp0L7BIL-I5RngVN3QP1gktwJD2Db. The page displays the "Google accounts" logo and the text "The application MIT AppInventor Version 2 is requesting permission to access your Google Account." and "Please select an account that you would like to use." Below this is a list of accounts with a radio button selected for the first one. The page also includes a disclaimer: "Google is not affiliated with the contents of MIT AppInventor Version 2 or its owners. If you sign in, Google will share your email address with MIT AppInventor Version 2 but not your password or any other personal information." and buttons for "Allow", "No thanks", and "Sign in to another account". There is also a checkbox for "Remember this approval for the next 30 days" and a copyright notice: "©2011 Google - Google Home - Terms of Service - Privacy Policy - Help".



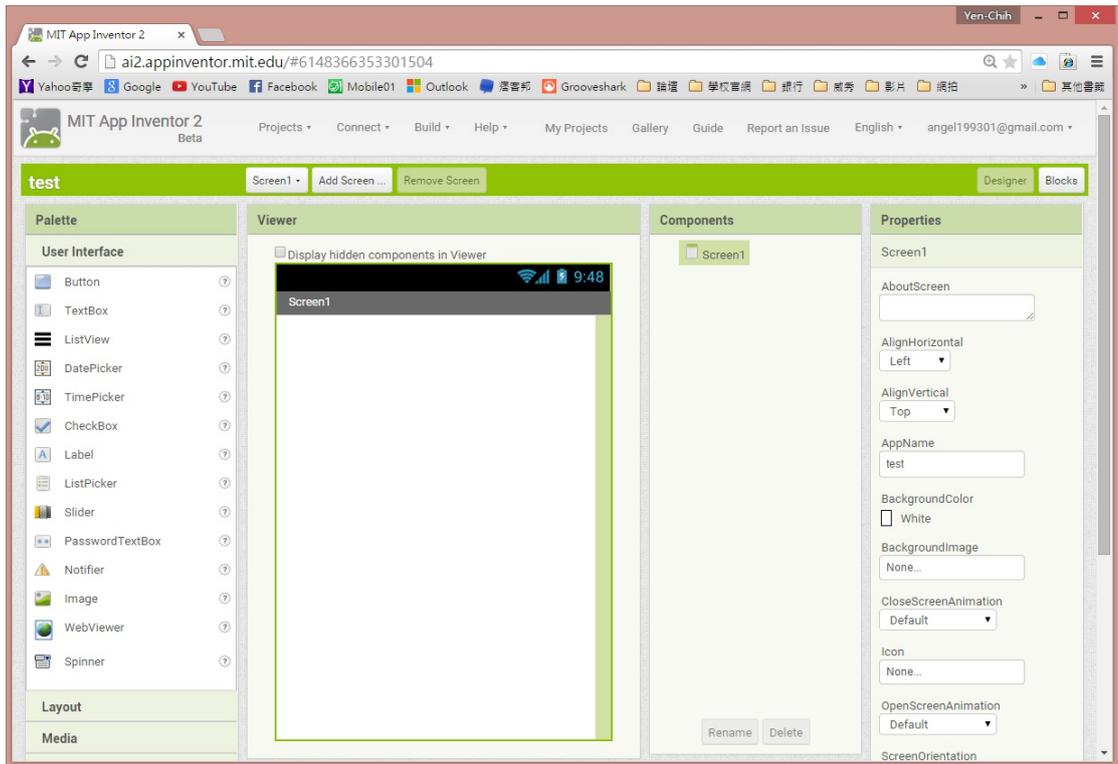
(2) 第一次使用先建立專案。



(3) 建立專案名稱



(4) 開始實作囉！



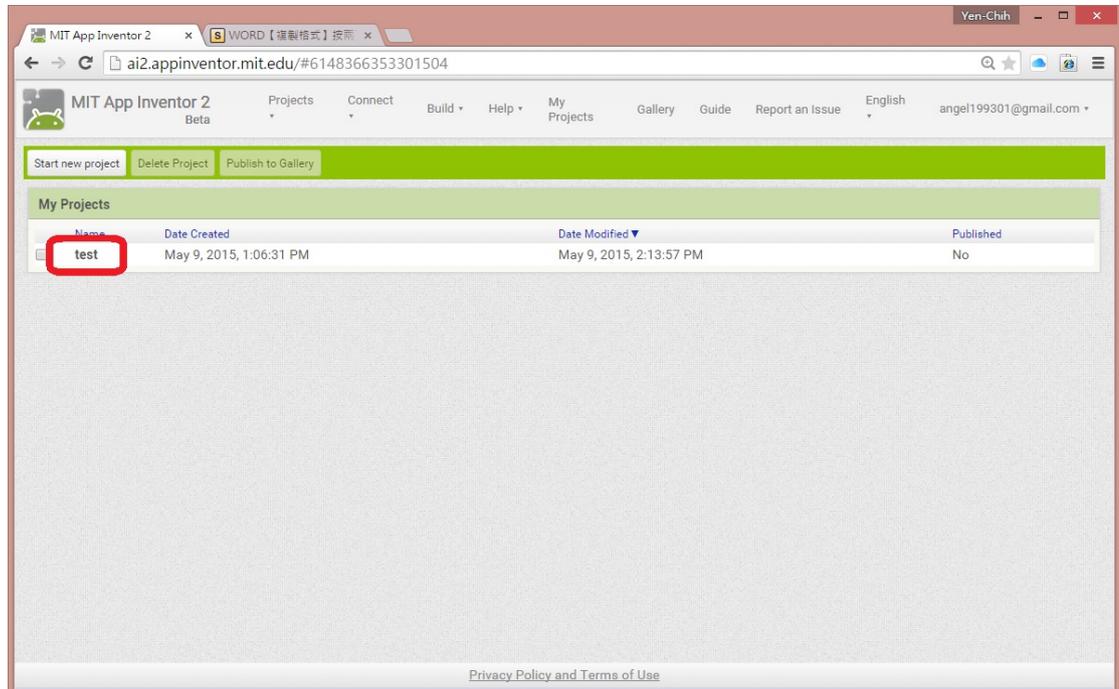
3. App Inventor 2 介面介紹

(1) 建立專案後會自動開啟版面配置頁面，方便使用者使用。左上會顯示專題名稱，左邊顯示元件區，中間顯示介面設計區，右方顯示放置以使用元件的 Components 區，按上方【My Projects】鈕可回到專題管理介面。



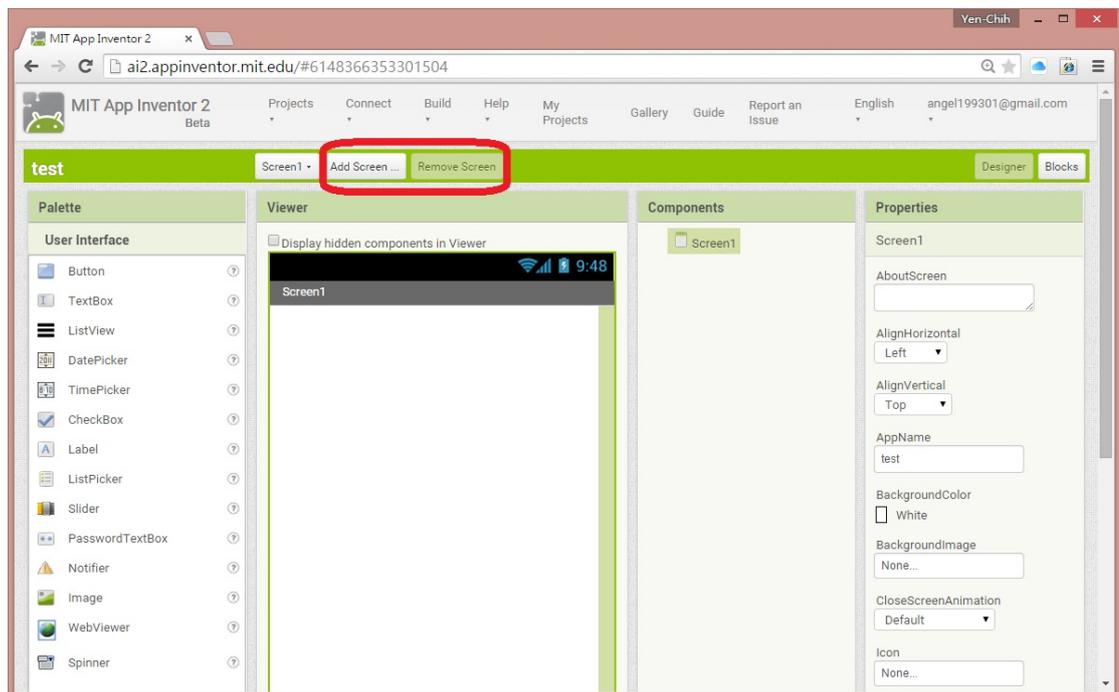
(2) 專題管理介面顯示所有已建立的專案。點選專案名稱可進入

版面配置頁面。(左方紅圈處)

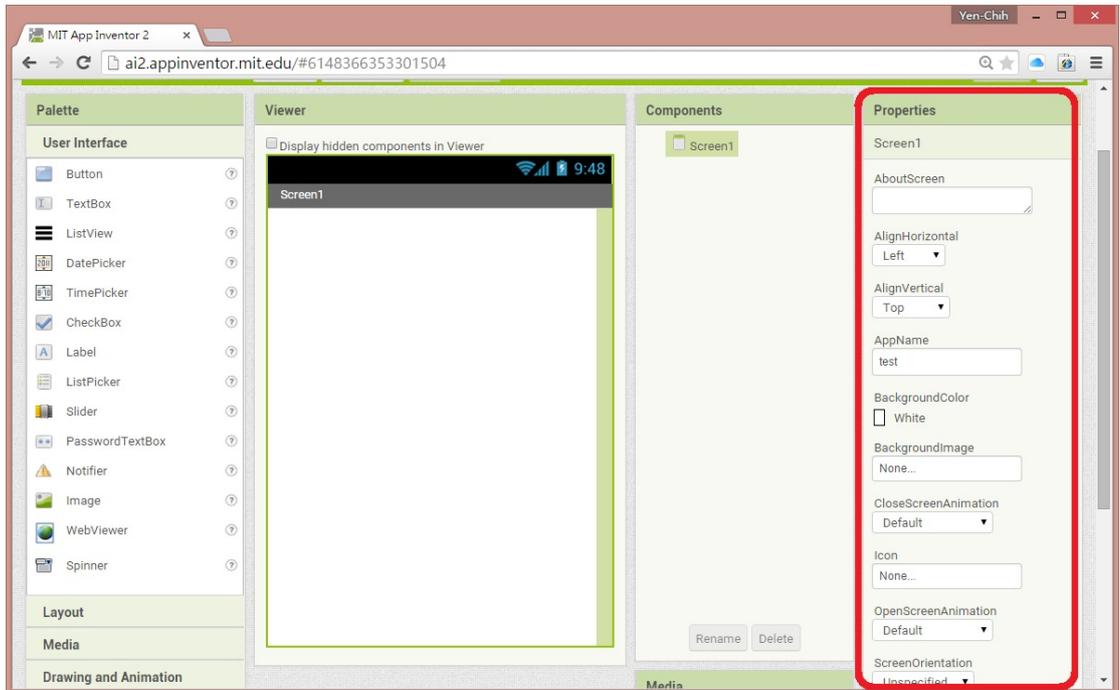


4. App Inventor 2 Screen 元件

(1) App Inventor 2 使用 Screen 元件產生應用程式頁面，一個 Screen 元件就是一頁。建立專題時，系統會自動為專案加入一個 Screen 元件，命名為 Screen1 設計者可以在此 Screen1 頁面中加入其他元件。



(2) 在已使用元件區右方是屬於設定區，當設計者點選元件後，屬性設定區會顯示該元件的屬性。系統會為大部分屬性設定預設值，設計者可以改變屬性值。

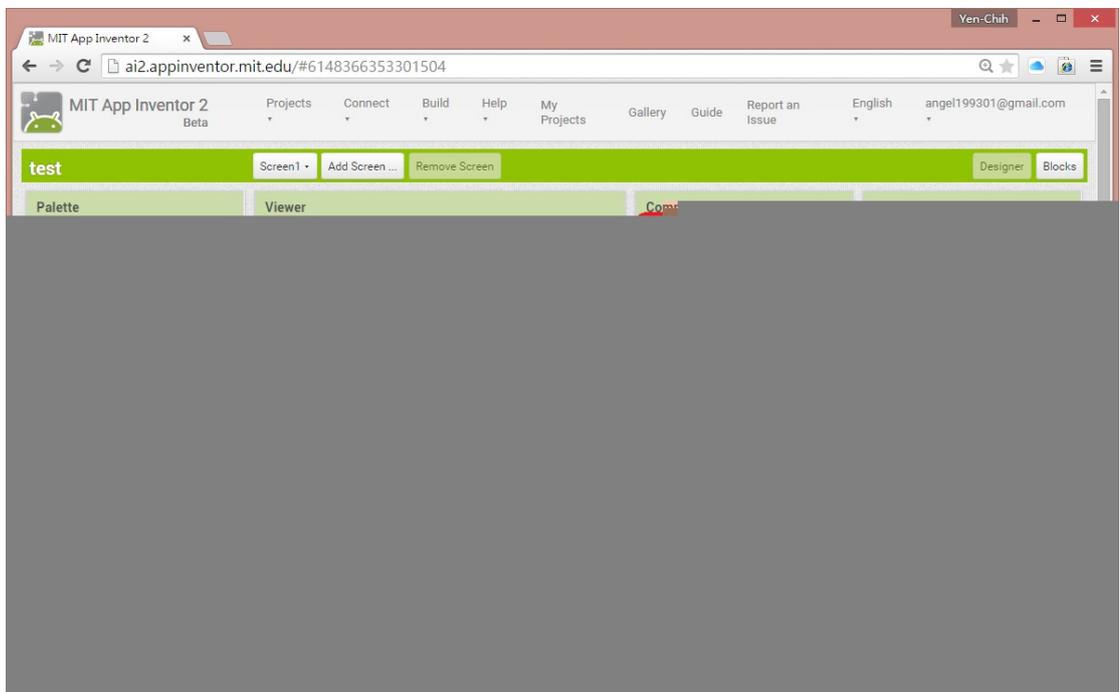


(3) Screen 的常用屬性有：

屬性	說明
AboutScreen	設定本頁面說明文字。
AlignHorizontal	設定水平方向對齊方式
AlignVertical	設定垂直方向對齊方式
BackgroundColor	設定背景顏色。
BackgroundImage	設定背景圖片。
CloseScreenAnimation	設定頁面關閉時呈現的動畫方式。
Icon	設定應用程式圖示，格視為 png 或 jpg，解析度 48x48。
OpenScreenAnimation	設定頁面開啟時呈現的動畫方式。
ScreenOrientation	設定螢幕方向，可能有三種屬性值： Unspecified：螢幕會隨手持方向自動改變 Landscape：螢幕固定以橫向顯示。 Portrait：螢幕固定以直向顯示。
Scrollable	設定當應用程式超過螢幕高度時，是否可以捲動。
Title	設定螢幕標題，顯示於螢幕的左上角，預設值「Screen1」

5. App Inventor 2 Lable 元件

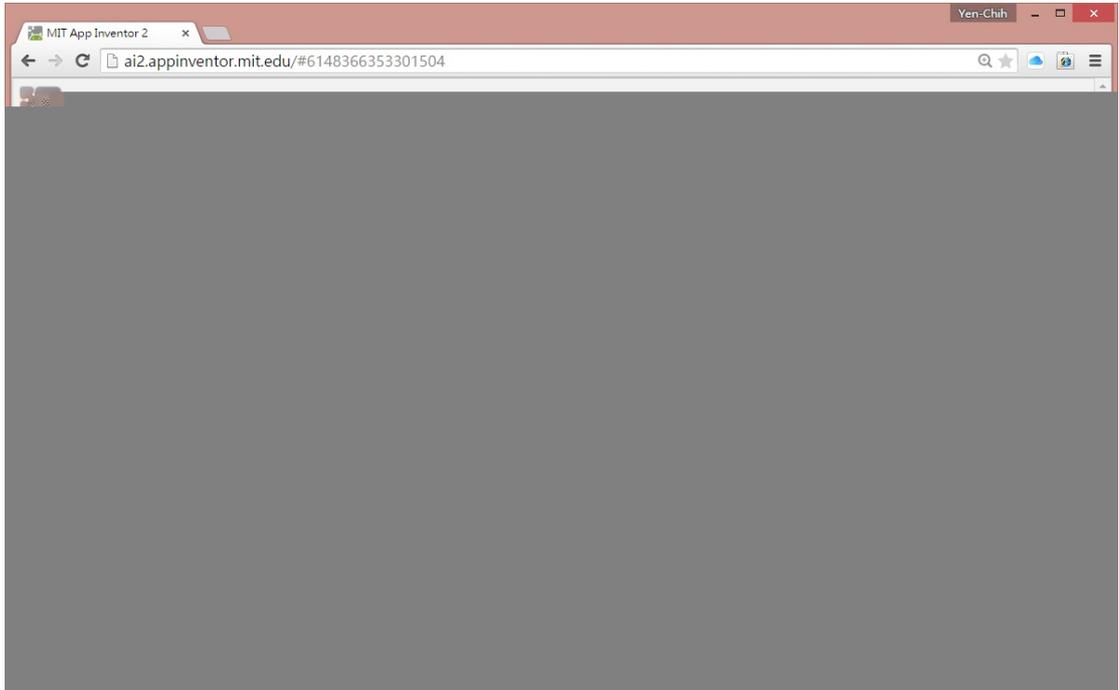
(1)Lable 元件是 App Inventor 2 使用最多的元件，其用途是顯示文字。App Inventor 2 中建立元件的方法非常簡單，只要將元件區的元件拖曳到介面設計區即可，例如建立一個 Lable 元件：由元件區拖曳 Lable 元件到介面設計區，在已使用元件區會多一個名稱為 Lable1 的 Lable 元件。



6. 拼塊編輯頁面

(1) 當應用程式的介面設定完成後，就可以切換到拼塊編輯頁面

〈Blocks〉，他最主要的功能是以拼塊來設計程式。



i. Designer 鈕：切換到版面配置頁面

ii. Blocks 鈕：切換到拼塊編輯頁面。

iii. 內建拼塊區：本區提供程式流程所需的程式拼塊。

iv. 已建立元件區：在版面配置頁面建立的元件，會在本區建立對應的元件拼塊。

v. 進階功能區：提供通用元件。

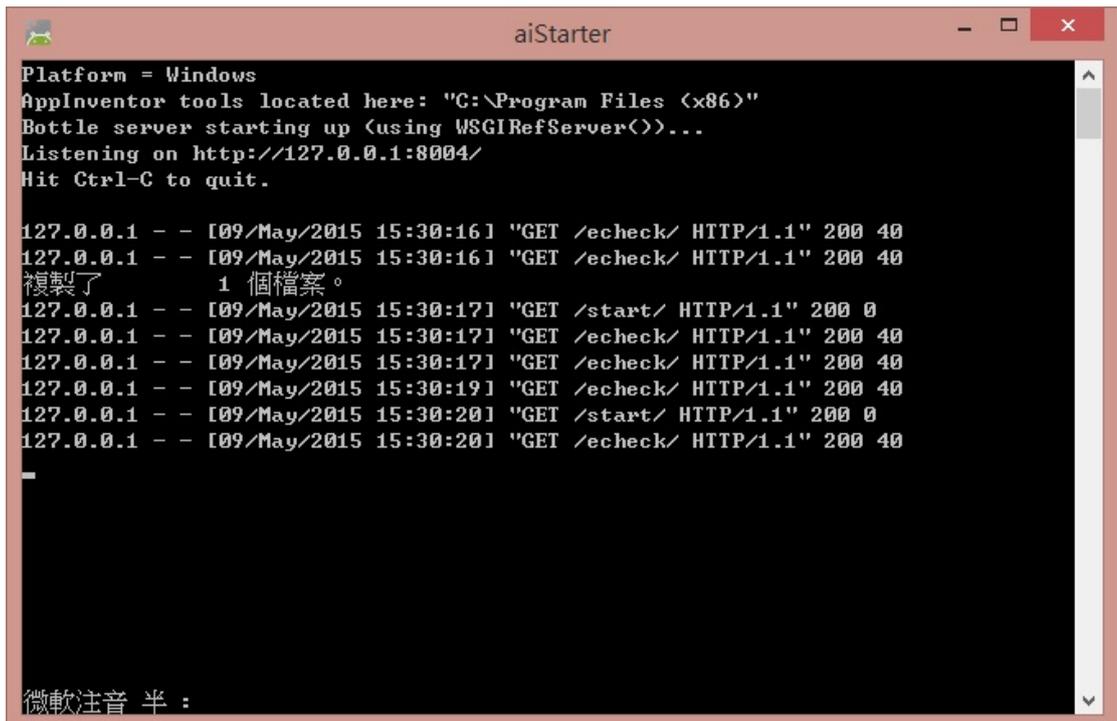
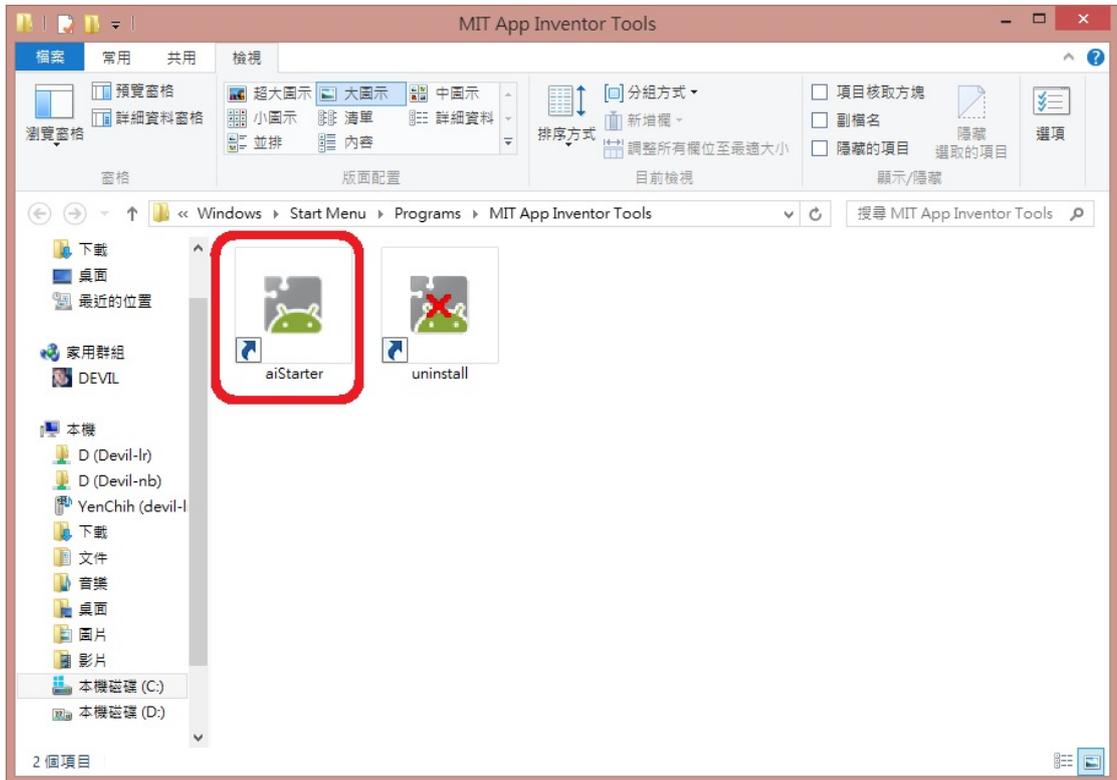
vi. 拼塊編輯區：可將各種拼塊由拼塊區拖曳到此區進行程式流程設計。

vii. 垃圾桶：將拼塊拖曳到垃圾桶可移除該程式拼塊。

7. 建立模擬器

(1) 安裝 APP Inventor2 時，系統已自動建立 aiStarter，再把它

啟動即可。

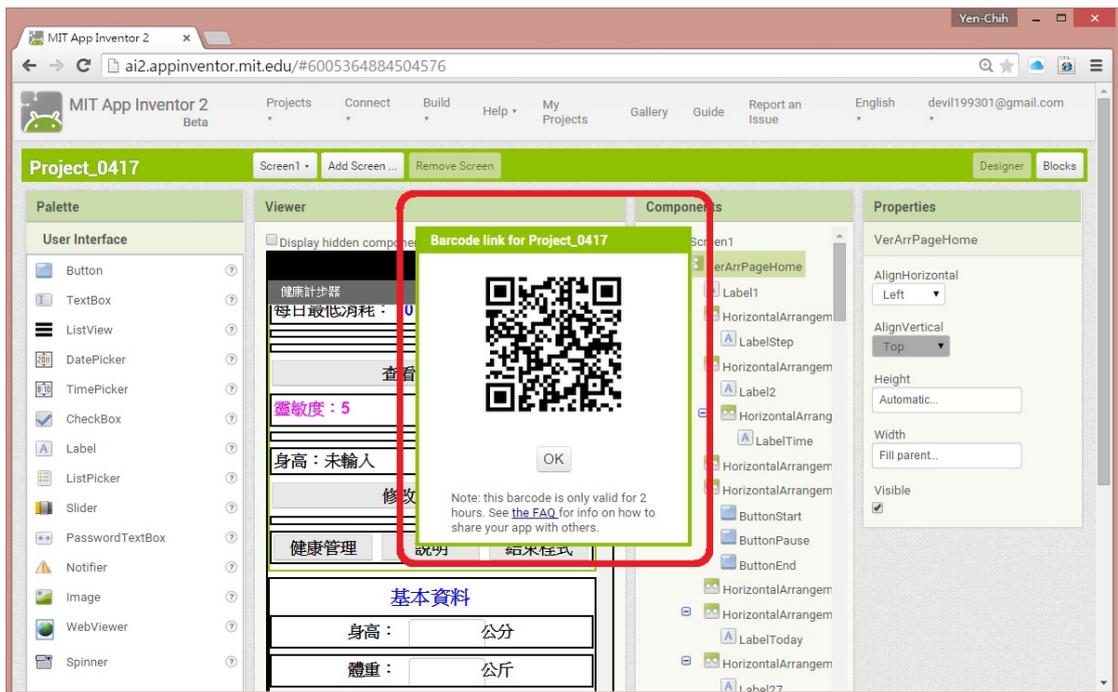
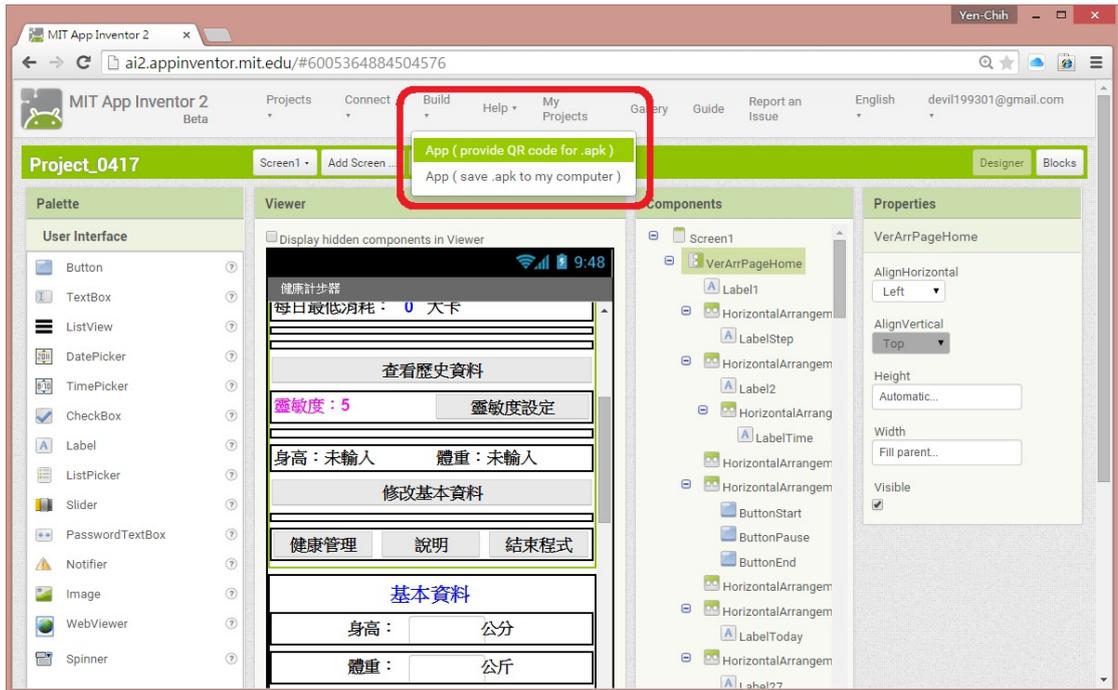


8. 在實機上執行

(1) 點選上方【Build】→【APP (provide QR code for .apk)】就

會產生 QR code，再把測試的手機連上網路後掃描 QR code

就可以安裝測試了。



第三章、系統操作介面

一、開啟

安裝完軟體之後，APP 頁面就會有一個軟體名稱叫「健康管理計步器」(如圖 3-1.1)。



(圖 3-1.1)

二、第一次使用必須先設定基本資料

當開啟 APP 時，會先讀取資料庫裡面的資料，當資料庫裡面沒有資料時，就會要求使用者先輸入基本資料(如圖 3-2.1)，然後自動轉換到輸入基本資料的頁面(如圖 3-2.2)。



(圖 3-2.1)

(圖 3-2.2)

三、主畫面顯示使用者輸入的資訊

在輸入完基本資料後回到 APP 主頁，中間就會顯示剛剛所輸入的資料(如圖 3-3.1)。



(圖 3-3.1)

四、【健康管理】

從主畫面進入健康管理後，程式會讀取資料庫裡的體重，方便使用者使用(如圖 3-4.1)。當按下『健康計算機』就會計算出使用者的 BMI (身體質量指數)、BMR (基礎代謝率)、體格 (過輕、適中、過重) 及標準體重(如圖 3-4.2)，幫助使用者了解自己的體格是否健康。



(圖 3-4.1)

(圖 3-4.2)

五、【減重計畫】

在健康管理頁面中，還有一個很重要的功能就是【減重計畫】，當使用者想要減肥時，輸入「目標體重」以及「預計減肥天數」，計算機就會幫你算出使用者平均每天須而外消耗多少卡路里，為避免使用者過度減肥，所以在時間的部分最短為 30 天(如圖

3-5.1)。

當使用者設定減重計畫後，會將資料存入資料庫，之後開啟軟體就不用重新設定，並且會把需消耗卡路里顯示在軟體主畫面，方便使用者隨時查看(如圖 3-5.2)。



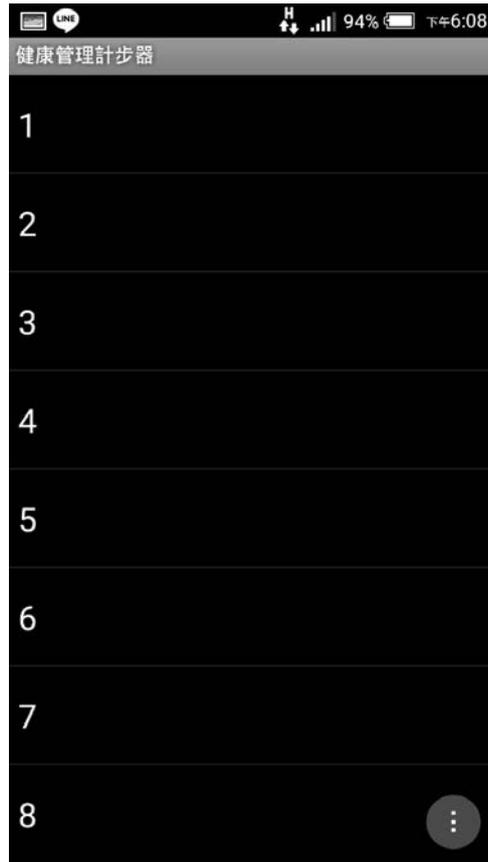
(圖 3-5.1)

(圖 3-5.2)

六、【計步器】前置設定

基本資料輸入完後，就可以使用計步器，但因為使用者的設備不一樣，靈敏度也就會不一樣，所以建議使用者在使用計步器之前

先設定靈敏度(如圖 3-6.1)，預設值為：5。



(圖 3-6.1)

七、【計步器】開始使用

靈敏度調整完就可以正常使用計步器了，按下「開始」就會計算時間以及行走步數（如圖 3-7.1）；按下「暫停」時間、計步器會暫停（如圖 3-7.2）；按下「停止」會顯示今日累積的步數以及消耗的熱量，並且將資料儲存至資料庫（如圖 3-7.3），當消耗的熱量大於減肥計畫中的最低熱量，就會告知使用者完成今日的減肥

目標（如圖 3-7.4）。

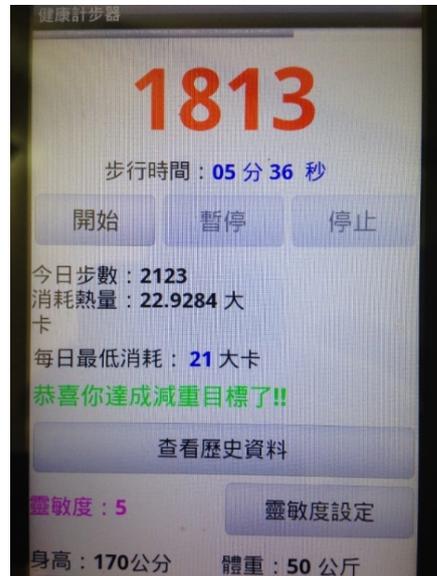


(圖 3-7.1)

(圖 3-7.2)



(圖 3-7.3)



(圖 3-7.4)

八、【查看歷史資料】

紀錄中會顯示過去六十天內每天行走的步數及消耗的熱量(如圖

3-8.1)，方便使用者好追蹤過去有沒有達成減肥的目標。



日期	步數	消耗熱量(大卡)
20150528	15	0.162

回主頁面

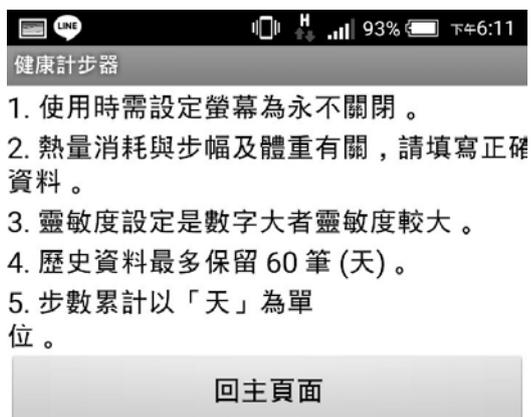


(圖 3-8.1)

九、【說明】

在軟體首頁下方有個「說明」，進入之後會顯示本軟體相關資訊

及注意事項(如圖 3-9.1)。



(圖 3-9.1)

第四章、完成目標與結論心得

(一)、完成目標

因應每個人不同的步行習慣，設計了 10 種不同靈敏度，使

用者可自行調整應用程式的靈敏度數值。

由於加速度感測器的靈敏度非常高，要控制加速度感測器非常難，我們是每隔指定時間才讀取加速度感測器的值。

一般行動裝置為節省電力，通常會在一段時間自動關閉螢幕，而加速度感測器在螢幕關閉時，會自動降低靈敏度，導致測的步行數遠比實際步行少，專題執行時，行動裝置螢幕一定要全程開啟，為避免使用者忘記設定永不關閉螢幕，應用程式執行時會顯

示提示訊息

(二)、未來展望

這次的專題研究，我們的重點主要是陀螺儀的運用，因為陀螺儀已經越來越熱門了，隨著手機的進步，陀螺儀幾乎可以完成所有的事情，藉著這一次的專題製作，可以讓我們去了解陀螺儀的運用原理。

第六章、參考文獻

Android 介紹：<http://zh.wikipedia.org/wiki/Android>

App Inventor 2 介紹：<http://www.appinventor.tw/ai2>

App Inventor 2 中文學習網：<http://www.appinventor.tw/>

BMI：

<http://www.dreye-health.com/SingularController?service=goCheckList&path=BMI>

BMR：

<http://www.dreye-health.com/SingularController?service=goCheck&path=bmr>

標準體重參考：

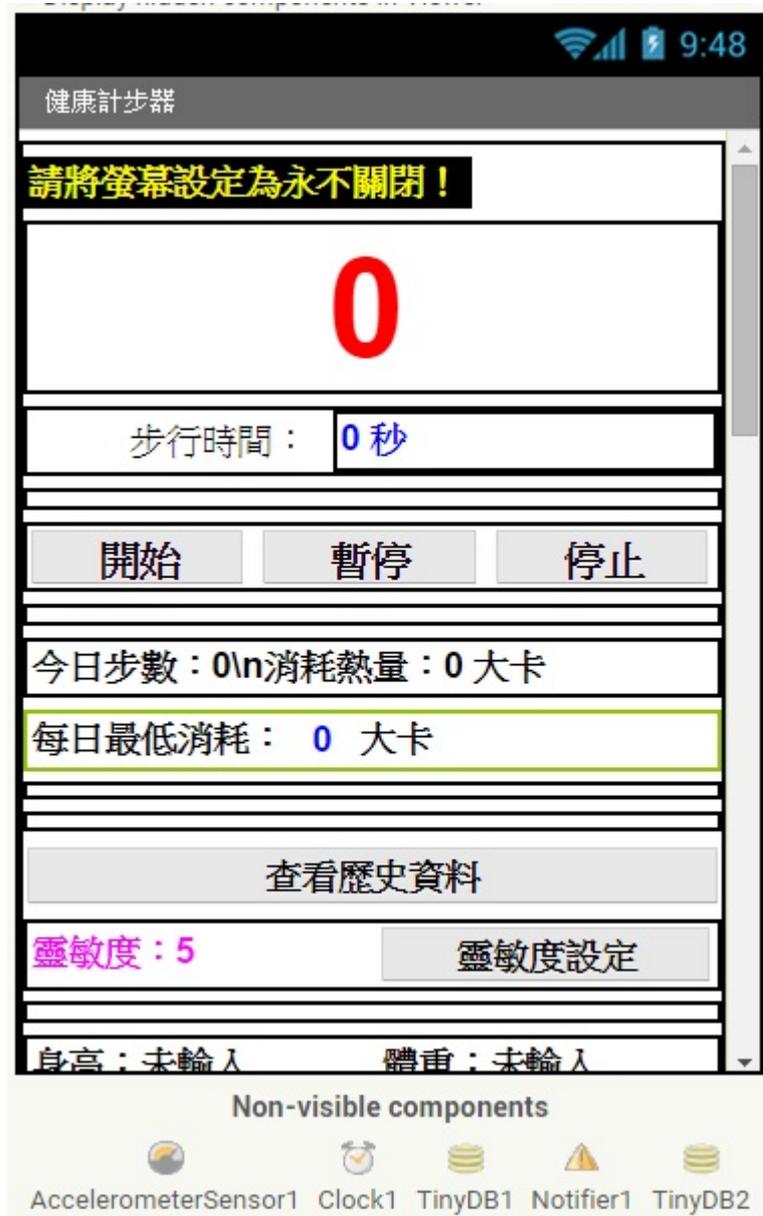
<http://www.scpo.nccu.edu.tw/show/part1/b/B2/standard.htm>

卡路里消耗計算：

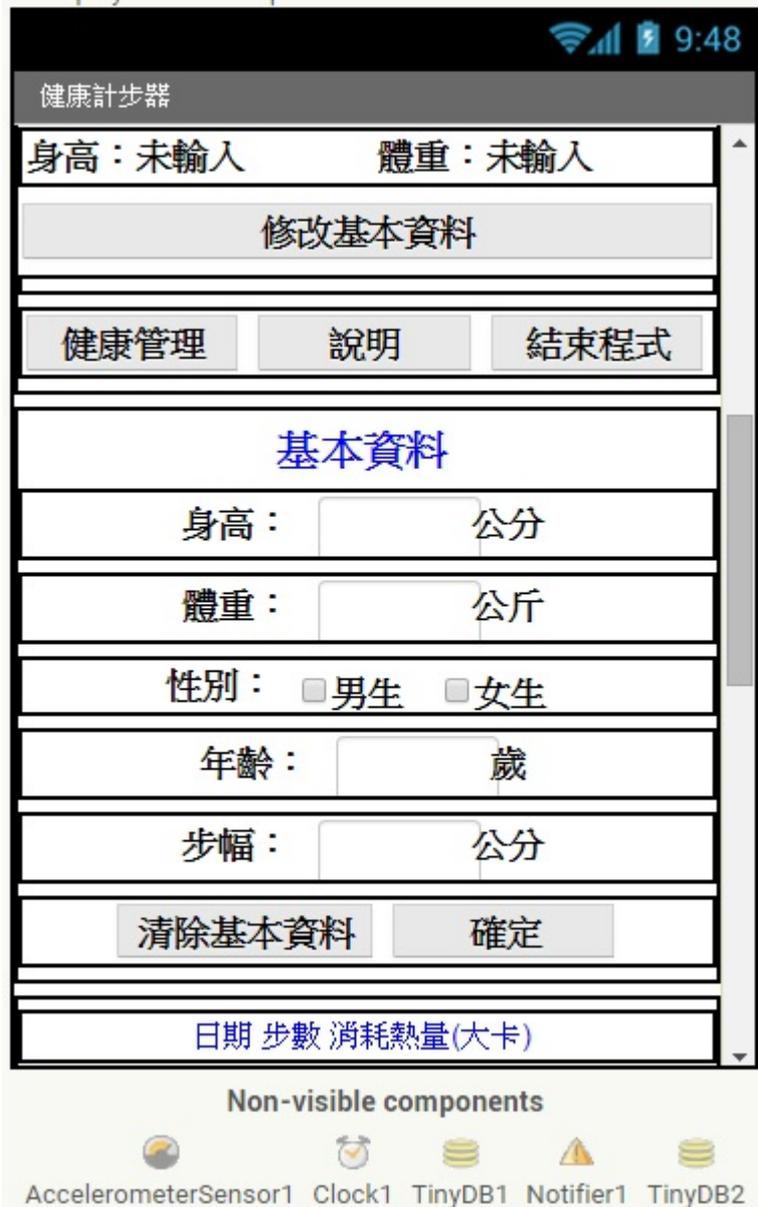
<http://www.scpo.nccu.edu.tw/show/part1/b/B2/bmr.htm>

附錄

(一)、頁面配置



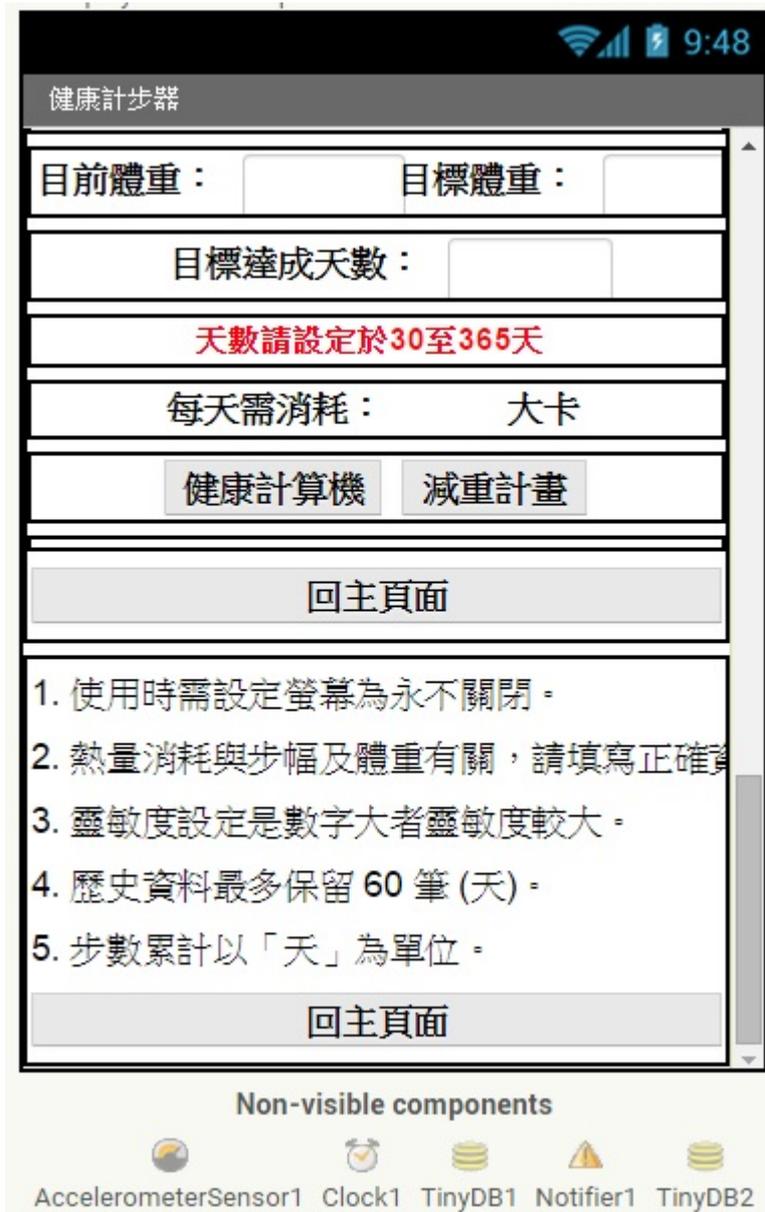
(版面配置-1)



(版面配置-2)



(版面配置-3)



(版面配置-4)

(二)、系統程式

1. 定義全域變數。

The image displays 26 'initialize global' blocks arranged in two columns. Each block consists of an orange header 'initialize global', a variable name, and a value or action. The variables and their values are:

- date to " "
- sy to 0
- firstDataDate to " "
- smart to 1
- sz to 0
- time to 0
- smartvar to 1.5
- timeHour to 0
- timeMin to 0
- smartstep to 0.1
- smartLevel to 5
- smartLevel to 5
- timeSec to 0
- start to false
- timeEnd to 0
- step to 0
- heat to 0
- timeInterval to 120
- maxRecord to 60
- timeStep to 0
- timeText to " "
- numTwoDigit to " "
- temNum to 0
- temText to " "
- stepList to create empty list
- stepList to create empty list
- temList to create empty list
- body to 0

2. 程式開始執行先取得系統時間、靈敏度資料及顯示今日步行數及讀取之前所使用的紀錄，例如：基本資料、減肥計畫。

```

when Screen1.Initialize
do
  set global times to call (Clock) Now
  if is empty call (myDB1) .GetValue
    tag smartData
    valueIfTagNotThere #
  then call (myDB1) .StoreValue
    tag smartData
    valueToStore get global smartLevel
  else set global smartLevel to call (myDB1) .GetValue
    tag smartData
    valueIfTagNotThere #
  set global smart to get global smartvar * get global smartstep * get global smartLevel
  set LabelSmart .Text to join 總步數: get global smartLevel
  if is a list? thing call (myDB1) .GetValue
    tag basicdataList
    valueIfTagNotThere #
  then set global basicdataList to call (myDB1) .GetValue
    tag basicdataList
    valueIfTagNotThere #
    set TextBoxLength .Text to select list item list get global basicdataList
      index 1
    set TextBoxWeight .Text to select list item list get global basicdataList
      index 2
    set TextBoxHeight .Text to select list item list get global basicdataList
      index 3
    set global basicdataList2 to call (myDB1) .GetValue
      tag basicdataList2
      valueIfTagNotThere #
    set TextBoxAge .Text to select list item list get global basicdataList2
      index 1
    set LabelWeight .Text to join 體重: TextBoxWeight .Text
    set LabelHeight .Text to join 身高: TextBoxHeight .Text
    set global sex to call (myDB1) .GetValue
      tag sex
      valueIfTagNotThere #
    set global aaa to call (myDB1) .GetValue
      tag aaa
      valueIfTagNotThere #
    set Label28 .Text to get global aaa
    set LabelTotal .Text to get global aaa
    if get global sex == 男性
    then set CheckBox1 .Checked to true
    if get global sex == 女生
    then set CheckBox2 .Checked to true
    if is a list? thing call (myDB1) .GetValue
      tag stepList
      valueIfTagNotThere #
    then call getDate
    call getFirstDate
    if get global firstDate == get global date
    then set global temNum to select list item list get global temList
      index 2
      * TextBoxWeight .Text
    set global heat to get global temNum * TextBoxLength .Text / 100 * 0.00072
    set LabelToday .Text to join 今日步數: select list item list get global temList
      index 2
      消耗熱量: get global heat
      大小
    while true
    if length of list list get global stepList >= get global maxRecord
    do remove list item list get global stepList
      index get global maxRecord + 1
    call (myDB1) .StoreValue
      tag stepList
      valueToStore get global stepList
    call ScreenClear
    set VisiblePageHome .Visible to true
    call buttonEnable
      TFbuttonStart true
      TFbuttonPause false
      TFbuttonEnd false
      TFbuttonHistory true
      TFbuttonSmart true
      TFbuttonInput true
    else call (Notifier) .ShowChooseDialog
      message 目前輸入步數資料開始使用計步器!
      title 輸入步數
      button1Text 輸入資料
      button2Text #
      cancelable false
  
```

3. 自訂程序 getDate 可取得今天的日期字串。

```
to getDate
do
  set global date to call Clock1 .Year
  instant get global time
  call twoDigit
  num call Clock1 .Month
  instant get global time
  set global date to join get global date
  get global numTwoDigit
  call twoDigit
  num call Clock1 .DayOfMonth
  instant get global time
  set global date to join get global date
  get global numTwoDigit
```

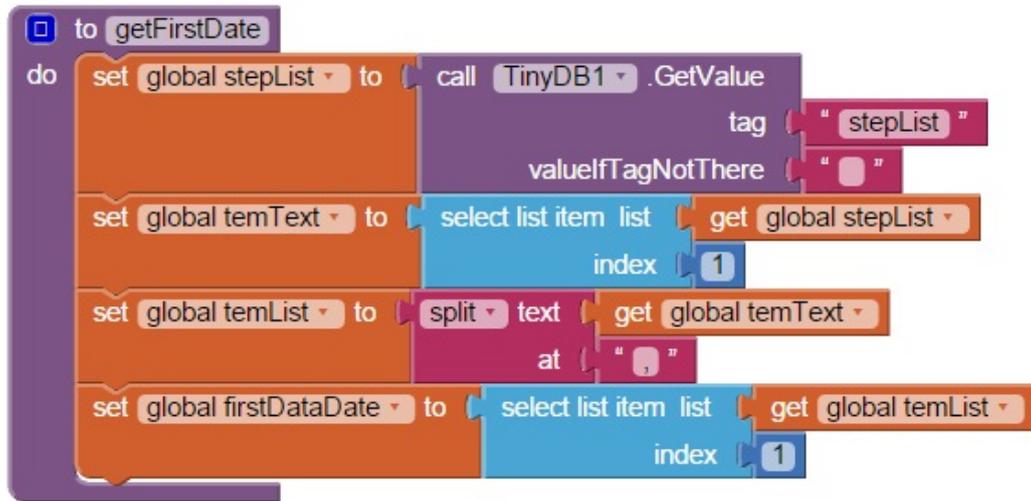
4. 自訂程序twoDigits可將一位數或二位數的數值轉換為兩個字元的字串，若是一位數就在數值前面加一個「0」。



```
to twoDigit num
do
  if (get num < 10)
  then
    set global numTwoDigit to join ("0" get num)
  else
    set global numTwoDigit to get num
```

The image shows a Scratch code block for a function named 'twoDigit' that takes a parameter 'num'. The function uses an 'if' statement to check if 'num' is less than 10. If true, it concatenates '0' and 'num' into a string and sets it to a global variable 'numTwoDigit'. If false, it sets 'numTwoDigit' to the value of 'num'.

5. 自訂程序getFirstDate可取得步行資料中第一筆資料的日期。



```
to getFirstDate
do
  set global stepList to call TinyDB1 .GetValue
  tag "stepList"
  valueIfTagNotThere ""
  set global temText to select list item list
  get global stepList
  index 1
  set global temList to split text
  get global temText
  at ""
  set global firstDataDate to select list item list
  get global temList
  index 1
```

The code defines a function named `getFirstDate`. It begins with a `do` block containing four `set` blocks. The first `set` block sets `global stepList` to the result of `call TinyDB1 .GetValue`. The second `set` block sets `global temText` to `select list item list` with `get global stepList` as the list and `index 1` as the index. The third `set` block sets `global temList` to `split text` with `get global temText` as the text and `at ""` as the separator. The fourth `set` block sets `global firstDataDate` to `select list item list` with `get global temList` as the list and `index 1` as the index.

6. 自訂程序ScreenClear會隱藏所有頁面，要達到此目的，只需將所有頁面的Visible 屬性值都設定為false 就行了！



```
to ScreenClear
do
  set VerArrPageHome . Visible to false
  set VerArrPageInput . Visible to false
  set VerArrPageShow . Visible to false
  set VerArrHelp . Visible to false
  set VerArrPageHealth . Visible to false
```

The image shows a Scratch code block for a function named 'ScreenClear'. The function is enclosed in a purple 'do' block. Inside, there are five 'set' blocks, each setting the 'Visible' property of a specific array element to 'false'. The array elements are VerArrPageHome, VerArrPageInput, VerArrPageShow, VerArrHelp, and VerArrPageHealth.

7. 本專題開始、暫停、停止、查看歷史資料及靈敏度設定五個按鈕，常需視情況改變其是否可以作用，因此將其寫成自訂程序，以五個按鈕的作用情況作為參數，只要執行此程序就能同時設定五個
- 按鈕

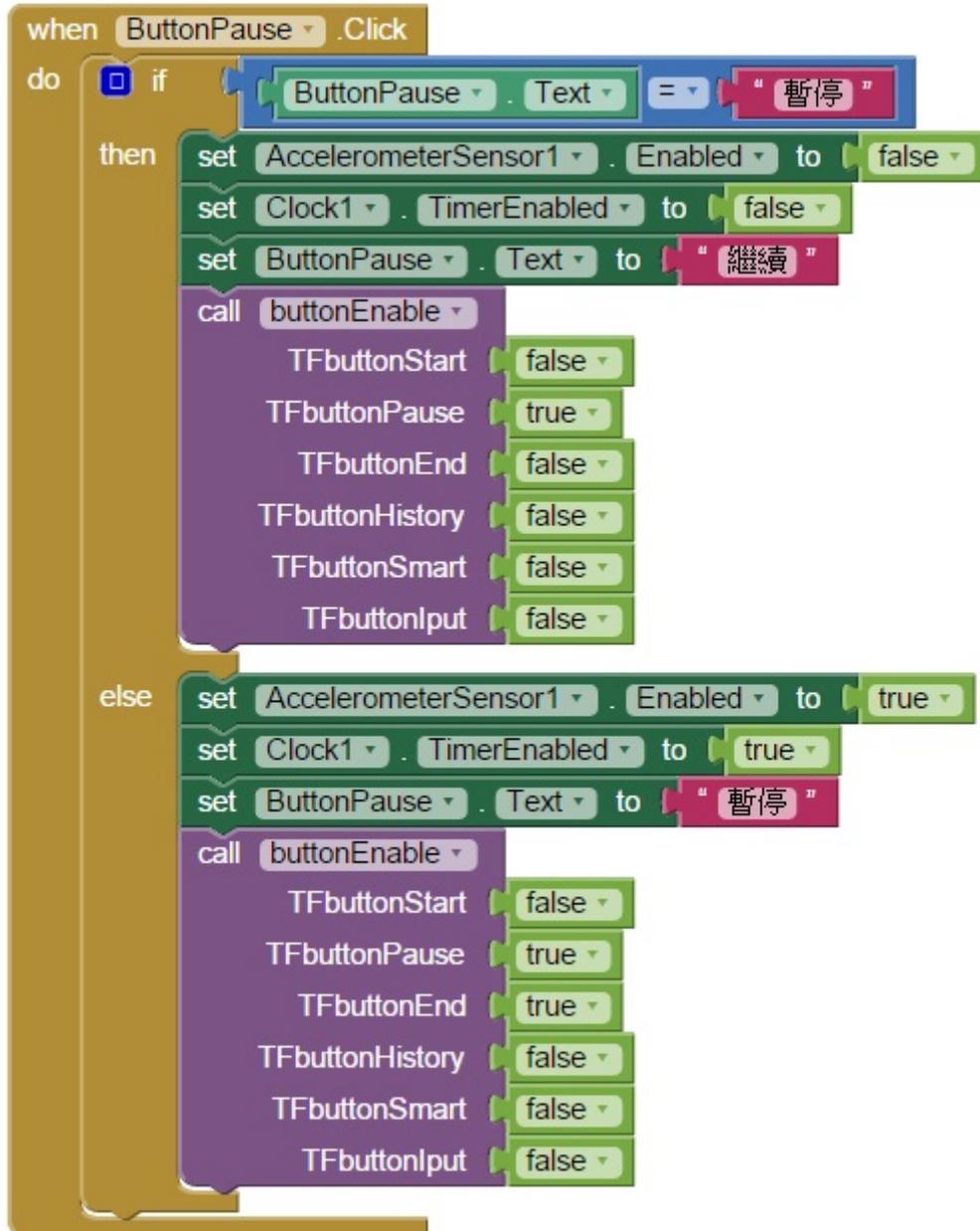
```
to buttonEnable TFbuttonStart TFbuttonPause TFbuttonEnd TFbuttonHistory TFbuttonSmart TFbuttonInput
do
  set ButtonStart . Enabled to get TFbuttonStart
  set ButtonPause . Enabled to get TFbuttonPause
  set ButtonEnd . Enabled to get TFbuttonEnd
  set ButtonHistory . Enabled to get TFbuttonHistory
  set ListPickerSmart . Enabled to get TFbuttonSmart
  set ButtonInput . Enabled to get TFbuttonInput
```

8. 當使用者按開始鈕就開始計算步行數:將步行數及步行時間歸零、
啟動計時器及加速度感測器。

```
when ButtonStart .Click
do
  set global step to 0
  set global timeStep to 0
  set LabelStep .Text to "0"
  set LabelTime .Text to "0"
  set Clock1 .TimerEnabled to true
  set global timeStart to call Clock1 .SystemTime
  set AccelerometerSensor1 .Enabled to true
  call buttonEnable
    TFbuttonStart false
    TFbuttonPause true
    TFbuttonEnd true
    TFbuttonHistory false
    TFbuttonSmart false
    TFbuttonInput false
```

The image shows a Scratch code block starting with a 'when ButtonStart .Click' event. Inside a 'do' loop, it performs several actions: sets 'global step' and 'global timeStep' to 0; sets the text of 'LabelStep' and 'LabelTime' to '0'; enables 'Clock1' and 'AccelerometerSensor1'; sets 'global timeStart' to the current system time using 'Clock1 .SystemTime'; and calls a 'buttonEnable' function. The 'buttonEnable' function is a call block with seven arguments: TFbuttonStart (false), TFbuttonPause (true), TFbuttonEnd (true), TFbuttonHistory (false), TFbuttonSmart (false), and TFbuttonInput (false).

9. 暫停鈕有兩個功能：如果按鈕文字為「暫停」，按下會停止計步，同時按鈕文字會變為「繼續」；若按鈕文字為「繼續」，按下會繼續計步，同時按鈕文字會變為「暫停」。



10. 使用者按停止鈕就將步行數記錄於資料庫，若有設定減肥計畫，

消耗卡路里大於設定值，顯示「已達成減重目標」。

```
when ButtonEnd .Click
do
  set AccelerometerSensor1 . Enabled to false
  set Clock1 . TimerEnabled to false
  set global start to false
  set ButtonPause . Text to "暫停"
  call getDate
  if is a list? thing call TinyDB1 .GetValue
    tag "stepList"
    valueIfTagNotThere ""
  then call getFirstDate
  else
    set global temNum to get global step
    set global temText to join
      get global date
      ""
      get global temNum
    add items to list list
      get global stepList
      item get global temText
  call TinyDB1 .StoreValue
    tag "stepList"
    valueToStore get global stepList
  set global heat to call calHeat
    num2 get global temNum
  set LabelToday . Text to join
    "今日步數："
    get global temNum
    "\n;消耗熱量："
    get global heat
    "大卡"
  if get global aaa > 0
  then
    if get global heat ≥ get global aaa
    then set Label30 . Text to "恭喜您已達成今日減重目標了!"
  call buttonEnable
    TFbuttonStart true
    TFbuttonPause false
    TFbuttonEnd false
    TFbuttonHistory true
    TFbuttonSmart true
    TFbuttonInput true
```

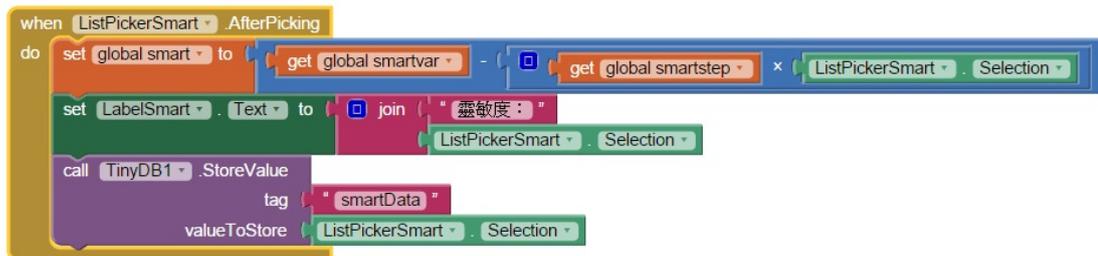
11. 計算步行時間：Clock1 元件的 TimerInterval 屬性值為預設的 1000，即每一秒會觸發一次 Clock1.Timer 事件，所以在此事件中將秒數加 1 就可正確計時。為了美觀，本專題採「xx 小時 xx 分 xx 秒」格式顯示，若數值為零則不顯示，例如步行 5 分 8 秒，顯示為「05 分 08 秒」。

```
when Clock1.Timer do
  set global timeStep to (get global timeStep + 1)
  set global timeSec to remainder of (get global timeStep ÷ 60)
  set global timeMin to quotient of (remainder of (get global timeStep ÷ 3600) ÷ 60)
  set global timeHour to quotient of (get global timeStep ÷ 3600)
  set global timeText to ""
  if (get global timeHour > 0) then
    call twoDigit num (get global timeHour)
    set global timeText to (join (get global timeText (get global numTwoDigit) "小時"))
  if (get global timeMin > 0) then
    call twoDigit num (get global timeMin)
    set global timeText to (join (get global timeText (get global numTwoDigit) "分"))
  call twoDigit num (get global timeSec)
  set global timeText to (join (get global timeText (get global numTwoDigit) "秒"))
  set LabelTime.Text to (get global timeText)
```

12. 使用者按查看歷史資料鈕會切換到歷史資料頁面，逐筆顯示步行數。

```
when ButtonHistory .Click
do
  set LabelHistory . Text to " "
  set global temText to " "
  for each var in list get global stepList
  do
    set global temList to split text get var
    at "."
    set global temText to join get global temText
    select list item list get global temList
    index 1
    call fiveStr
    num3 select list item list get global temList
    index 2
    set global temText to join get global temText
    get global numTwoDigit
    set global heat to call calHeat
    num2 select list item list get global temList
    index 2
    set global temText to join get global temText
    get global heat
    "\n"
  set LabelHistory . Text to get global temText
  call ScreenClear
  set VerArrPageShow . Visible to true
```

13. 靈敏度設定鈕是 ListPicker 元件，元件名稱為 ListPickerSmart1，其 ElementsFromString 屬性值設定為「1,2,3,4,5,6,7,8,9,10」，這是 ListPicker 元件的選單項目值。當使用者按下靈敏度設定鈕，系統會顯示 1 到 10 十個項目供使用者選取，使用者點選任一個項目值後，就會觸發 ListPickerSmart1.AfterPicking 事件。ListPickerSmart1.AfterPicking 事件會將選取的靈敏度等級轉換為靈敏度數值(smart)，同時在主頁面顯示靈敏度等級，同時寫入資料庫。



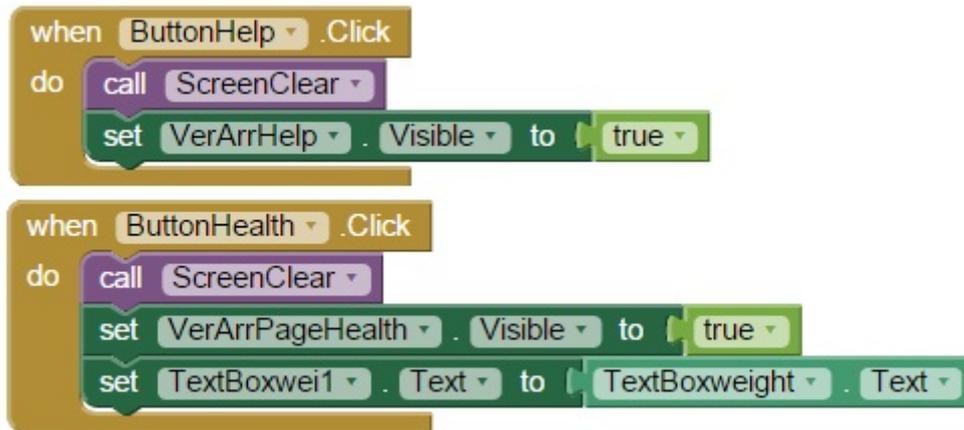
```
when ListPickerSmart1 .AfterPicking
do
  set global smart to (get global smartvar - (get global smartstep * ListPickerSmart1 . Selection))
  set LabelSmart1 .Text to (join ("靈敏度：" ListPickerSmart1 . Selection))
  call TinyDB1 .StoreValue
    tag "smartData"
    valueToStore ListPickerSmart1 . Selection
```

14. 當使用者按結束程式鈕，就顯示對話方塊詢問使用者是否確定要
結束程式。



15. 當使用者按說明鈕就切換到使用說明頁面，當使用者按健康管理

就切換到健康管理頁面。



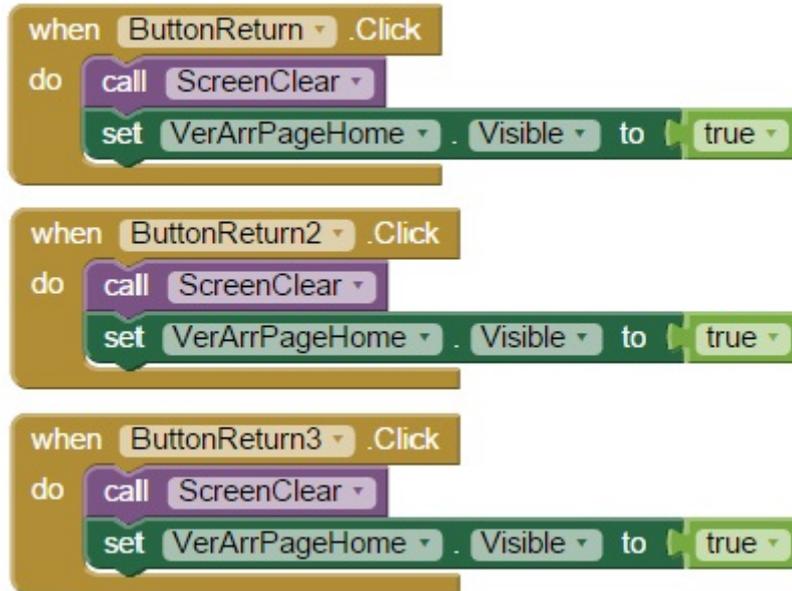
```
when ButtonHelp .Click
do
  call ScreenClear
  set VerArrHelp . Visible to true

when ButtonHealth .Click
do
  call ScreenClear
  set VerArrPageHealth . Visible to true
  set TextBoxwei1 . Text to TextBoxweight . Text
```

The image shows two Scratch code blocks. The first block is triggered by a 'when ButtonHelp .Click' event and contains two actions: 'call ScreenClear' and 'set VerArrHelp . Visible to true'. The second block is triggered by a 'when ButtonHealth .Click' event and contains three actions: 'call ScreenClear', 'set VerArrPageHealth . Visible to true', and 'set TextBoxwei1 . Text to TextBoxweight . Text'.

16. 在歷史資料頁面及使用說明頁面和健康計畫各有一個回主頁面
鈕(ButtonReturn 及 ButtonReturn2 及 ButtonReturn2)，按下後

會切換到主頁面。



```
when ButtonReturn .Click
do
  call ScreenClear
  set VerArrPageHome . Visible to true

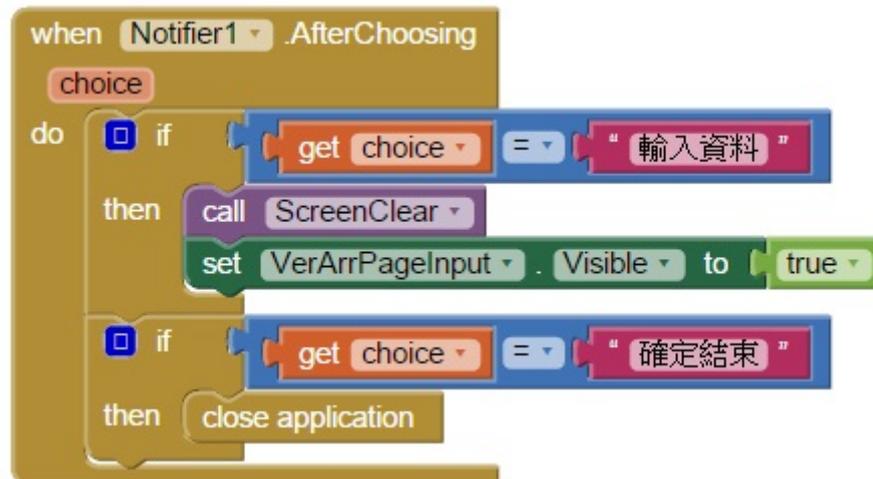
when ButtonReturn2 .Click
do
  call ScreenClear
  set VerArrPageHome . Visible to true

when ButtonReturn3 .Click
do
  call ScreenClear
  set VerArrPageHome . Visible to true
```

17. 在主頁面按結束程式鈕後，使用 Notifier 元件顯示的對話方塊，

在對話方塊中按確定結束鈕就以 `close.application` 結束應用程

式。

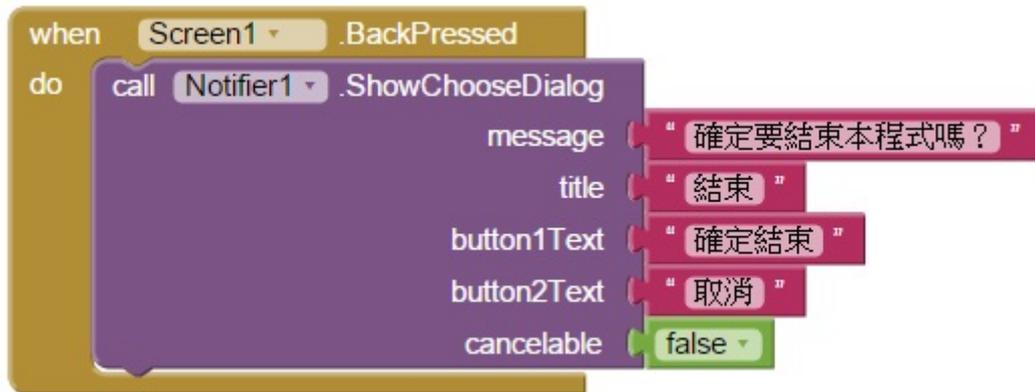


18. 本專題是震動時才要計次，而震動時三軸數值必然會改變，所以使用三軸數值的差來偵測手機震動。其特色是三個方向的數值改變都要偵測：因為不同使用者計步時手機擺放的方向及位置可能不同，因此無法得知加速度感測器哪一個軸會產生變化，所以三個軸都要偵測，只要任何一個軸改變都要計步。

```
when AccelerometerSensor1 AccelerationChanged
  xAccel yAccel zAccel
  do
    set global timeEnd to call Clock1 SystemTime
    if
      get global timeEnd - get global timeStart > get global timeInterval
    then
      if
        get global sz = false
      then
        set global start to true
      else
        if
          absolute get xAccel - get global sx > get global smart or
          absolute get yAccel - get global sy > get global smart or
          absolute get zAccel - get global sz > get global smart
        then
          set global step to get global step + 1
          set LabelStep Text to get global step
          set global timeStart to call Clock1 SystemTime
          set global sx to get xAccel
          set global sy to get yAccel
          set global sz to get zAccel
```

19. 按行動裝置上的返回鍵(「<」)，會彈出確認結束應用程式對話

方塊，作用與在主頁面按結束程式鈕相同。



20. 減重計畫：包含判斷體重、日期輸入值是否正確，然後計算出減

重平均每天必須消耗多少卡路里。

```

when Buttonvwv.Click
do
  if length(TextBoxwei1.Text) <= 0 or length(TextBoxwei2.Text) <= 0 or TextBoxwei1.Text <= TextBoxwei2.Text or TextBoxwei1.Text <= 0
  then
    call Notifier.ShowChooseDialog
    message 您的【體重】或【天數】輸入錯誤呢!
    title 減重計畫
    button1Text OK
    button2Text 
    cancelable false
  else
    if get global bmr <= 0
    then
      if CheckBox2.Checked
      then
        set global bmr to (1.8 * TextBoxheight.Text + 9.6 * TextBoxweight.Text + 655) / (4.7 * TextBoxage.Text)
      if CheckBox1.Checked
      then
        set global bmr to (13.7 * TextBoxweight.Text + 5 * TextBoxheight.Text + 66) / (6.8 * TextBoxage.Text)
      set global aaa to round((TextBoxwei1.Text - TextBoxwei2.Text) * 7700 / TextBoxday.Text)
      call TinyDB1.StoreValue
      tag aaa
      valueToStore get global aaa
      set Labelbmr.Text to join(BMR: , round(get global bmr))
      set Labeltotal.Text to get global aaa + get global bmr
      set Label25.Text to 減重每天需消耗熱量必須再加上飲食所增加的熱量，才能達到減重的目的囉!
      set Label28.Text to get global aaa
    
```

21. BMI、BMR、標準體重計算

The image shows a Scratch script for calculating BMI and BMR. The script starts with a 'when ButtonBMI Click' event. It then checks if 'CheckBox2' is checked. If so, it calculates BMR using the formula: $1.8 \times \text{TextBoxheight} \cdot \text{Text} + 9.6 \times \text{TextBoxweight} \cdot \text{Text} + 655 - 4.7 \times \text{TextBoxage} \cdot \text{Text}$. If 'CheckBox1' is checked, it calculates BMR using the formula: $13.7 \times \text{TextBoxweight} \cdot \text{Text} + 5 \times \text{TextBoxheight} \cdot \text{Text} + 66 - 6.8 \times \text{TextBoxage} \cdot \text{Text}$. Both paths set 'Labelbmr' text to 'BMR: ' followed by the rounded BMR value. Then, it calculates BMI using the formula: $\frac{\text{TextBoxweight} \cdot \text{Text}}{\frac{\text{TextBoxheight} \cdot \text{Text}}{100}^2}$. It sets 'LabelBMI2' text to 'BMI: ' followed by the rounded BMI value. The script then uses a series of 'if' statements to determine the BMI category based on the rounded BMI value:

- If BMI < 18.5, set 'Labelbody' text to '體重過輕' (Underweight).
- If BMI is between 18.5 and 24, set 'Labelbody' text to '體重適中' (Normal weight).
- If BMI is between 24 and 27, set 'Labelbody' text to '過重' (Overweight).
- If BMI is between 27 and 30, set 'Labelbody' text to '輕度肥胖' (Mild obesity).
- If BMI is between 30 and 35, set 'Labelbody' text to '中度肥胖' (Moderate obesity).
- If BMI is 35 or higher, set 'Labelbody' text to '重度肥胖' (Severe obesity).

Finally, it calculates the standard weight using the formula: $22 \times \frac{\text{TextBoxheight} \cdot \text{Text}}{100}^2$. It sets 'Labelbody2' text to '標準體重: ' followed by the rounded standard weight value and the unit '公斤' (kg).

全文完