

基督教香港信義會信義中學

中三普通電腦科

探究

MIT App Inventor 2



姓名： _____

班別： _____ ()

目錄

第 1 課	MIT App Inventor 2 簡介	2
第 2 課	計算圓面積的應用程式	6
第 3 課	聰明的畢氏定理應用	10
第 4 課	日語翻譯器	15
第 5 課	猜時間遊戲	20
專案	動態遊戲：地球保衛戰	28

下載圖像及音頻檔案網址：

<http://221.126.228.2/download/f3ict/ict.html>

功課紀錄

日期	內容	分數 / 評級

第一課

MIT App Inventor 2 簡介

1.1 App Inventor 是甚麼？

App Inventor 是一個免費的網上平台讓使用者開發供 Andorid 電話使用的 APP (應用程式)。使用者只要把類似砌圖的程式方塊(program blocks)拼湊起來，無須輸入複雜的電腦指令，便可完成一個智能電話應用程式。使用者每次都必須使用 Chrome 登入以下網站方可製作 App Inventor app。

App Inventor 網址：<http://ai2.appinventor.mit.edu/>

App Inventor 伺服器會把你的工作儲存在它的雲端儲存裝置內。當然你也可以把你的程式碼下載至你的電腦內。當你完成編程後，你可以在你的智能電話(以 USB 連線)或模擬器(emulator)執行程式，進行測試。

1.2 安裝 App Inventor

雖然 App Inventor 是一個網上平台，但是你也需要在電腦內安裝以下的軟件才可開始編寫 app：

(a) App Inventor 安裝程式

請到以下的網址下載和安裝 App Inventor 安裝程式：

<http://appinventor.mit.edu/explore/content/windows.html>

(b) 你的電腦也需要配備最新版本的 Google Chrome 瀏覽器。注意，App Inventor 平台只能在 Chrome 上運作。

你也需要一個 Google 用戶戶口 (你可使用現有的用戶戶口) 登入 App Inventor 平台。

1.3 App Inventor 介面

要建立一個 app，我們需要

- 在 **Designer (畫面編排)** 頁面建立一個手機的畫面 (screen)
- 在 **Blocks Editor (程式設計)** 編製程式方塊
- 啟動軟件 **aiStart**，(它是把程式傳送到手機或模擬器的軟件)
- 點選 [Connect] → [Emulator] / [USB] 把你的 app 傳送到手機或模擬器進行程式測試。



活動 1 第一個由我編寫的 App

你將會登入 App Inventor 平台，並編寫一個簡單的 app，向使用者打個招呼。

1. 啟動 Chrome
2. 瀏覽網頁 <http://ai2.appinventor.mit.edu/>
3. 輸入你個人的 Google 戶口資料，登入 App Inventor 平台。
4. 登入後，你可看到 Designer(畫面編排) 介面上的一個空白的手機畫面 "Screen1"。
(如果 App Inventor 要求你輸入專案名稱，請輸入 **Greeting** 。)



5. 在 Palette(元件面板)視窗拉一個文字輸入盒和一個按鈕 "Screen1"。

問題

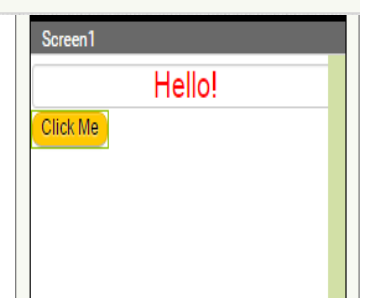
(a) 這個文字輸入盒叫甚麼名字？

(b) 這個按鈕叫甚麼名字？

6. 按一下文字輸入盒，並在「元件屬性」視窗中改變它的性質。
字體大小 = 30
寬度 = 填滿(Fill parent ...)
文字對齊= 置中 文字 = Hello!
文字顏色 = 紅色



7. 按一下按鈕，並更改它的屬性：
背景顏色 = 橙色 形狀 = 圓角
文字 = Click Me
8. 按一下 [程式設計] (Blocks) 按鈕，轉到程式設計介面。



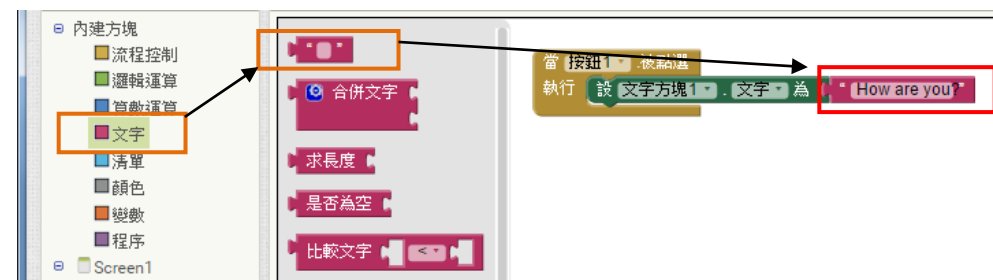
9. 按一下**按鈕 1** 圖標，在打開的方塊清單上拉出 "當按鈕 1.被點選" 方塊到工作面板(viewer)。



10. 按一下**文字輸入盒 1** 圖標，在打開的方塊清單上搜尋(下方)，拉出 " 設文字輸入盒 1.文字為 " 方塊到 "當按鈕 1.被點選" 方塊之中



11. 在內件方塊(Build-in section) 點一下「文字」，再在清單中拉出一個 " " 方塊，並把它接連至 "設文字輸入盒 1.文字為" 方塊的右端。在 " " 方塊輸入"How are you?". (如下圖示)



12. 執行 aiStart 軟件。
13. 儲存專案，然後點選 **連線(Connect) → 模擬器(Emulator)**。
14. 過了一兩分鐘後，電腦會出現智能電話的模擬器。
在模擬器上點選你的 app 圖標啟動程式。



問題 試簡單描述這個程式方塊的作用。

備註： 你可以到學校網站的「反轉教室」重溫本課的內容：

[電腦] → [中三] → [App Inventor - 第一課：第一個由我編寫的 App]

自主學習 - 筆記

1. App Inventor 是甚麼？

App Inventor 是 _____

2. App Inventor 的工作介面



在畫面編排介面，我們可以 _____

在程式設計介面，我們可以 _____

其他重點

第二課

計算圓面積的應用程式

2.1 認識數學函數

App Inventor 提供很多常用數學函數方塊用來建立應用程式，然而在 AI 使用這些數學函數方塊建立數學算式的方法有別於其他軟件，例如 Excel 和 Raptor。讓我們透過以下的活動學習如何應用 AI 的數學函數方塊吧。

課堂活動 1 認識 AI 的數學函數

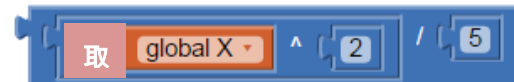
1. 識別以下的數學方塊，然後寫出對應的數學算式。（註：X 和 Y 是整數變量。）



例子： $8 - X$



$-(\quad + \quad)$



2. 參照右方的數學算式，在左方的空格內填上適當的內容，組成對應的程式方塊。

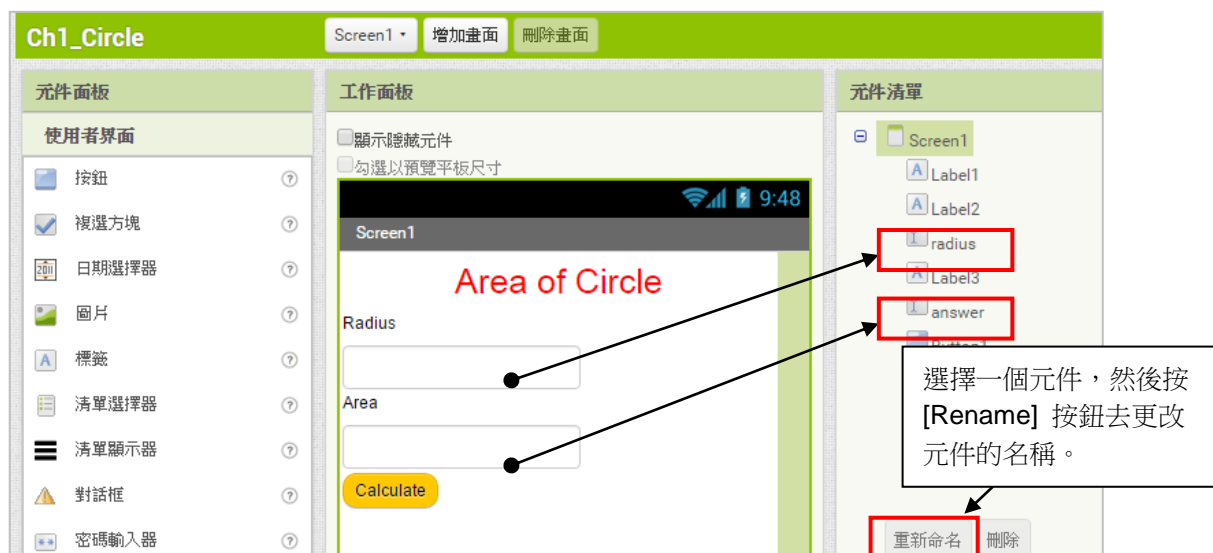
AI 程式方塊	數學算式
	$\sqrt{12}$
	$\cos(X)$

	$X \div 5$ 的餘數 (remainder)
	X 的四捨五入值
	$ Y $ (Y 的絕對值)

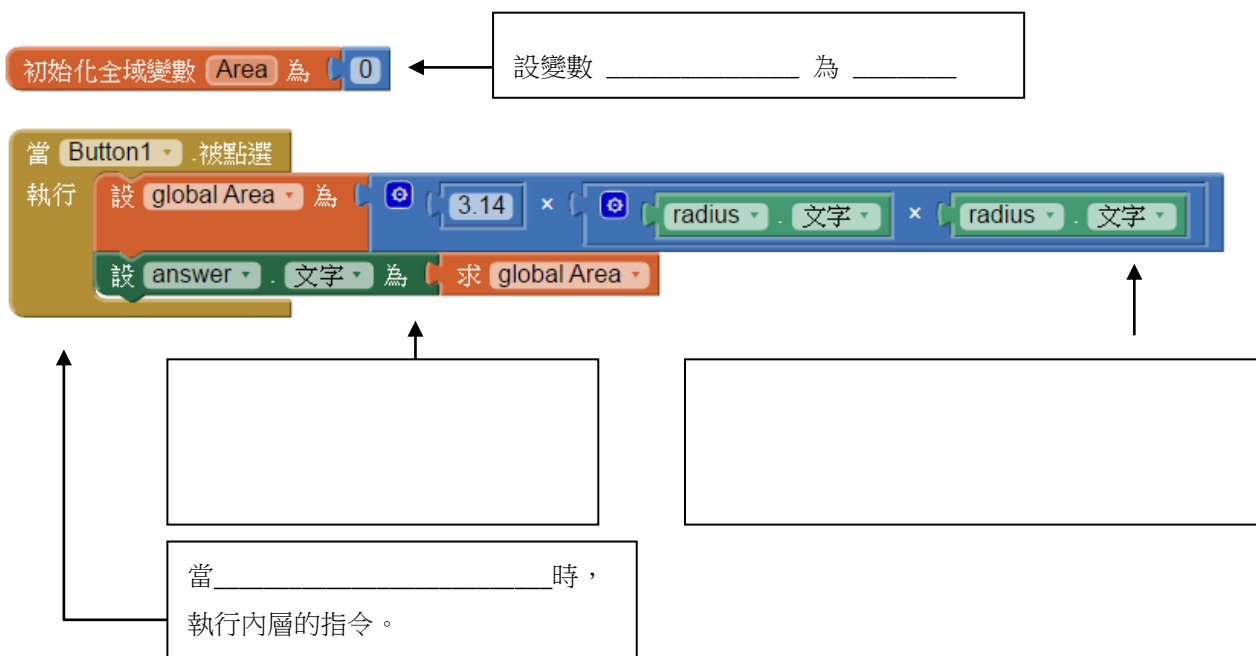
活動 2 編寫計算圓面積的應用程式

我們在這活動會編寫一個先讀取半徑數值然後計算它的圓面積的 AI 程式。

1. 在 Chrome 使用你的 Google 戶口登入 AI 網站 <http://ai2.appinventor.mit.edu/>
2. 開啟一個新的應用程式，並把它命名為 "circle_area"。
3. 在 **Designer (畫面編排)** 頁面，拖曳以下的元件至 "Screen1"，並按照以下的規格設定元件的格式，把兩個文字輸入盒更改命稱為 "radius" 和 "answer"。



4. 轉到[程式設計] **Blocks** 介面，建立以下程式方塊，並在空格內寫出這些方塊的功能。



5. 儲存你的專案(程式)。
6. 執行 **aiStart**。點選 **Connect → Emulator** 啟動模擬器和執行你的程式。
7. 使用以下的半徑數值測試你的程式。寫出電腦計出的面積。

(a) 半徑 = 5 面積 = _____ (b) 半徑 = 12.8 面積 = _____

課後練習 計算圓周的 App

修改活動 2 的應用程式，使它讀取使用者輸入的半徑值後，計算及顯示圓形面積和圓周。

(提示：你須在「畫面編排」(Design mode)介面上加入一個標籤和一個文字輸入盒。)

依照以下步驟列印你的 app。

1. 在「畫面編排」介面(Design mode) 按一下鍵盤上的 [Print Screen] 鍵，捕捉屏幕畫面。
2. 在 Word 文件內貼上畫面圖像。
3. 重複步驟 1 和 2 把[程式設計](**Blocks**) 介面的程式方塊畫面貼到 Word 文件內。
4. 調整兩幅圖像大小至一頁之內。
5. 列印 Word file 文件，剪下兩幅圖像，然後貼在下方的空格內。

提示：你可以觀看「反轉教室」的教學短片重溫本課及獲得本練習的進一步提示：

[電腦] → [中三] → [App Inventor - 第 2 課：計算圓周]

自主學習 – 編寫筆記

1. 使用者介面元件

標籤 是用作 _____

文字輸入盒 是用作 _____

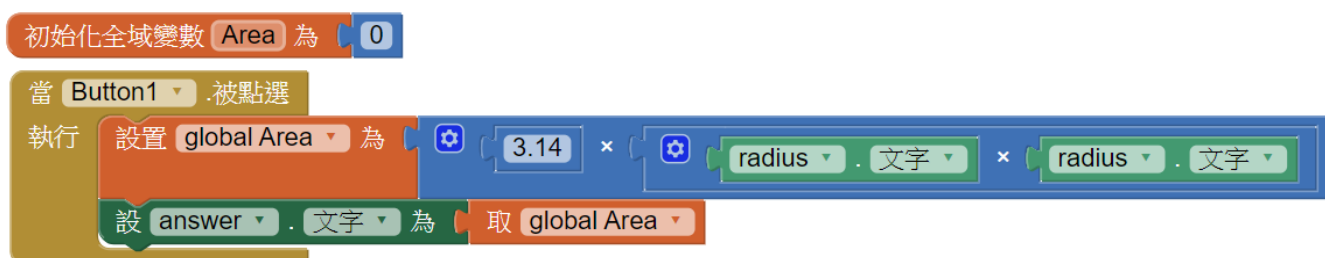
按鈕 是用作 _____

2. 方塊元件

在 App Inventor，變數(variable) 的作用是_____

在一個程式開始時，變數必須為_____

3. 分析程式方塊



在上方的程式方塊，

名為_____的變數開始時被 _____ 為數值 _____

當 "Button1" 被按下時，電腦會_____

其他重點筆記

第三課

聰明的畢氏定理

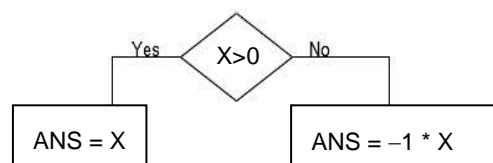
3.1 在 app 中插入圖像

智能電話應用程式常常使用到圖像、聲音和視像等多媒體元素。在 App Inventor，我們可以在「畫面編排」介面中使用"圖像"(image)元件，把一些靜態圖像(static images)加入至 app 的屏幕上。如果你要製作一個包含互動(interactive)圖像的應用程式，你便需要使用"繪圖動畫" (Drawing and Animation)的「圖像精靈」 (ImageSprite) 了。



3.2 控制方塊：IF ... then ... else

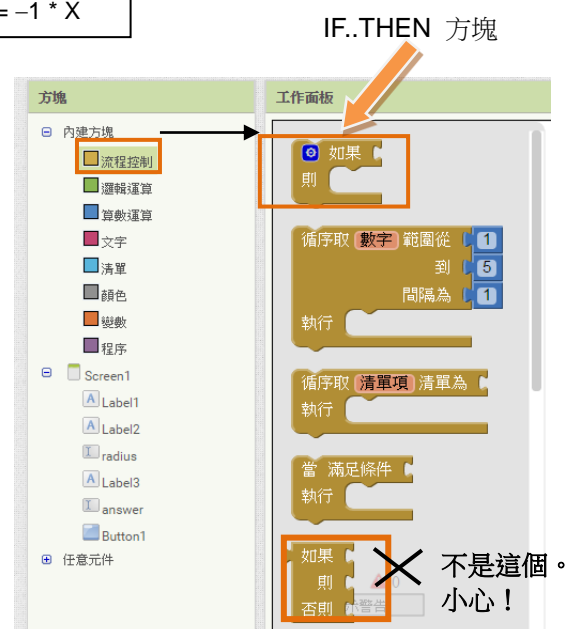
在 Raptor 編程的課程，我們已學會使用選擇符號 (selection symbol) 建立一個控制結構 (control structure)：



上方的 IF..THEN..ELSE 流程圖功能可用以下文字表示：

如果 X 是正數，則
 設變數 ANS 等於變數 X
否則
 設變數 ANS 等於 (-1 * X)

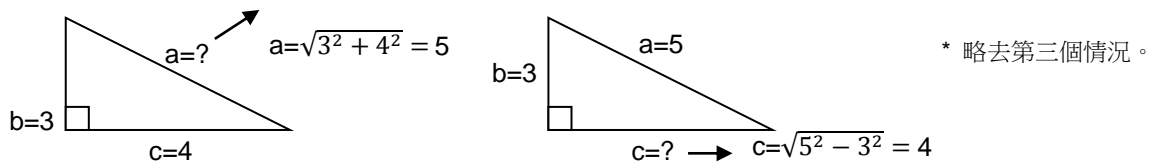
簡單而言，控制結構的作用是根據不用的輸入數值作出不同的輸出或反應。在 App Inventor，IF...THEN...的控制結構是採用 "內建" (Built-in) 部份的 "if...then" 方塊來表



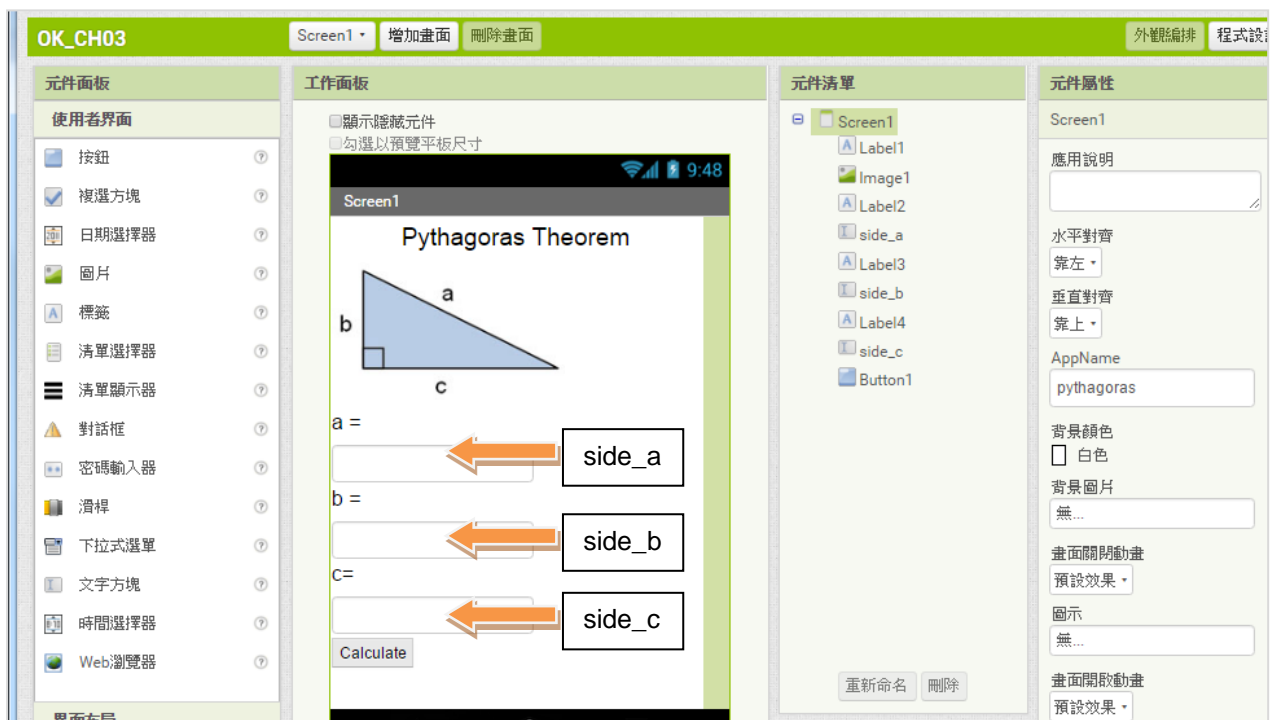
示的。

活動 3：畢氏定理 APP

我們在這活動建立一個應用程式，讀取任何兩條直角三角形的邊長，然後計算出第三條邊長。

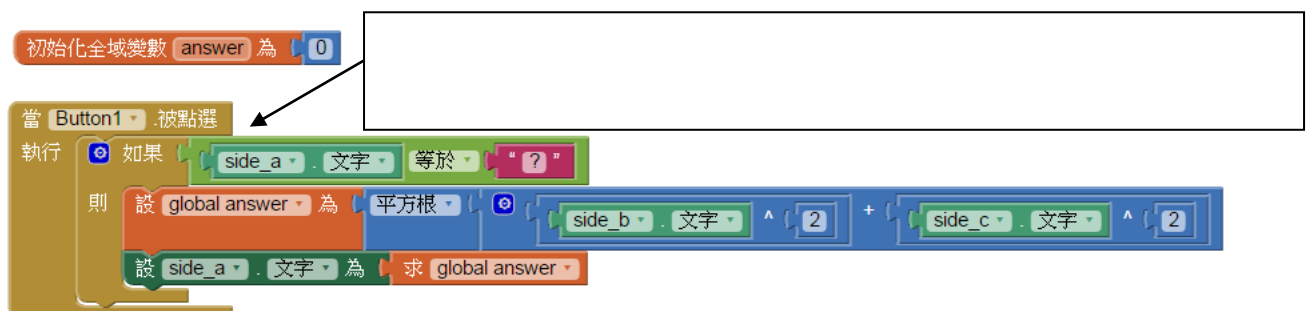


1. 開啟 Chrome，然後以你的 Google 戶口登入 AI 網站 <http://ai2.appinventor.mit.edu/>
2. 建立一個新的 app 專案，命名為 "pythagoras"。
3. 在「畫面編排」介面，把以下的元件加入"Screen1" 畫面中，並如下方般設定元件的格式。
重新命名畫面的三個文字輸入盒為 "side_a"、"side_b" 和 "side_c"。
(請在目錄頁所示的 Google 協作平台中下載三角形圖像 Ch03-pic02A.jpg。)



4. 轉到[程式設計] **Blocks** 介面，建立以下程式方塊。

試在右方空格中以文字描述 "when Button1.click" 和包含其內的程式方塊的功能。



- 複製整個 "if side_a.text = ..." 程式方塊，然後貼上一份副本，再修改其內容使它能够計算：

$$b = \sqrt{a^2 - c^2}$$

在以下程式方塊中的空格中填上適當的內容。



- 重複步驟 5，以製作程式方塊計算 $c = \sqrt{a^2 - b^2}$ 。
- 儲存你的專案。執行 `aiStarter` 去測試你的程式。
- 列印和在下面的空格貼上所有程式方塊。

自主學習 3

更多控制方塊的應用：計算兩個整數的最大公因數(HCF)

你還記得怎樣在 Raptor 流程圖找出兩個整數的 HCF？我們會在這個活動中觀看一個反轉教室的短片，按着指引建立一個計算兩個整數的 HCF 的應用程式。

1. 登入學校的反轉教室。
2. 按照以下的指引觀看短片：
[電腦] → [中三] → [App Inventor - 第 3 課：計算 HCF]
3. 按指引建立一個計算 HCF 的應用程式。
4. 列印你的程式方塊和把它貼在下面的空格內。

進階題目

1. 試解以下程式方塊的功能，並把執行完結時變數 X 和 Y 的值寫出來。



程式方塊的功能是 _____

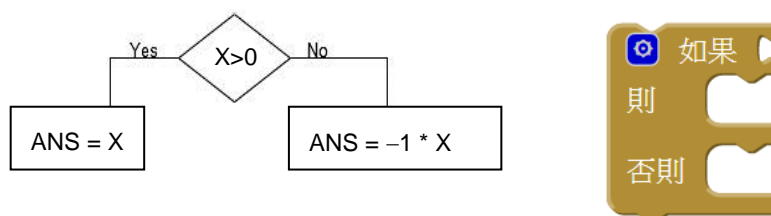
執行程式後，變數 X = _____ 變數 Y = _____

自主學習 – 編寫筆記

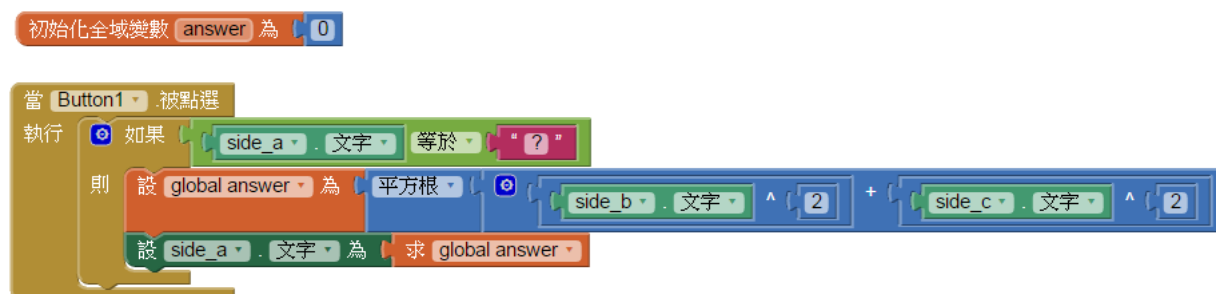
1. If ... Then ... Else ... 方塊

IF... Then... 程式方塊屬於「內件方塊」部份的 _____ 組別內。

在 App Inventor，以下的流程圖可以用 IF...THEN...ELSE 方塊表示：



2. 程式方塊分析



在以上的程式，變數 _____ 被初始化為_____。

當 Button1 被點選 _____

其他重點筆記

在清單裡，我們能夠輕鬆地管理大量數據（而無需建立大量不同名稱的變量元件）。

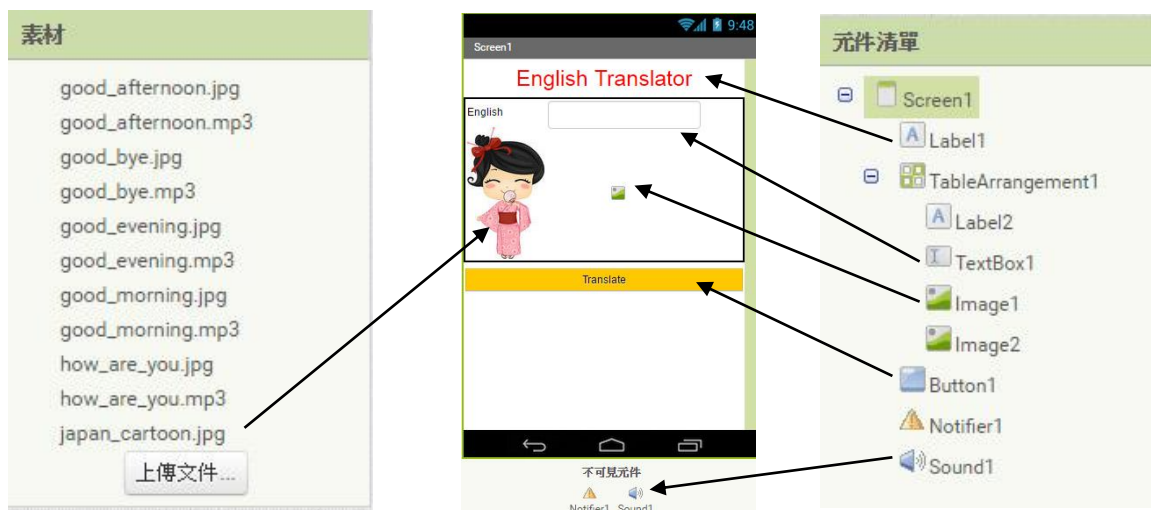
活動 4 隨身的日語翻譯員

在這活動，我們會建立一個日語翻譯的應用程式。它把使用者輸入的英文字句翻譯成日語，並讀出該日語。對於未能翻譯的語句(未能在清單"english"找到)，這應用程式會回應日語的 "我不知道。"

1. 在 Chrome 以你的 Google 戶口登入 AI 網站 <http://ai2.appinventor.mit.edu/>。
2. 建立一個新的專案 "translator"。
3. 在畫面編排介面，拉下一個標籤和一個「表格佈置」(在介面佈置) 至 "Screen1"。
設定 "TableArrangement1" 為一個 2 列和 2 行的佈局。



4. 上載以下的檔案至「素材」部份 (請先從 Google 協作平台下載至你的電腦)。
5. 加入其他元件，並設定其位置和格式(如右圖所示)。

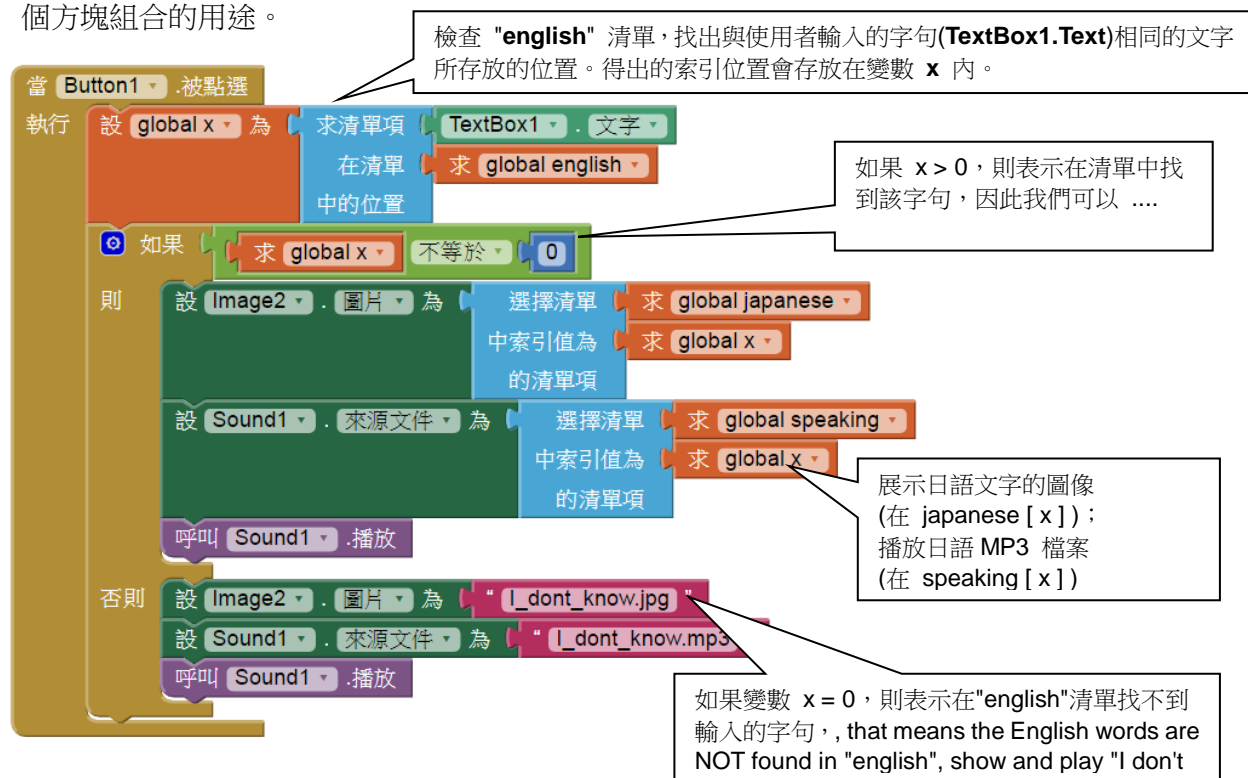


6. 轉到「程式設計」介面，建立一個變數 x 和三個清單變數(見右圖)。這三個清單的作用如下：

english	儲存英文字句
japanese	儲存日語文字圖像檔案的名稱
speaking	儲存日語 MP3 檔案的名稱



7. 建立 "when Button1.Click" 方塊和其包含的其他程式方塊。請用心閱讀下方的描述，學習每一個方塊組合的用途。



8. 儲存專案，然後執行 aiStarter 和測認你的程式。

延展活動 (反轉教室支援)

修改活動 4 的應用程式，加入四個額外的語句：

good bye , see you soon , you are beautiful , you are handsome

列印你的程式方塊，並把它們貼在下方的空格內。

課後功課

1. 寫出一個「清單」變數和一般變數的主要分別。

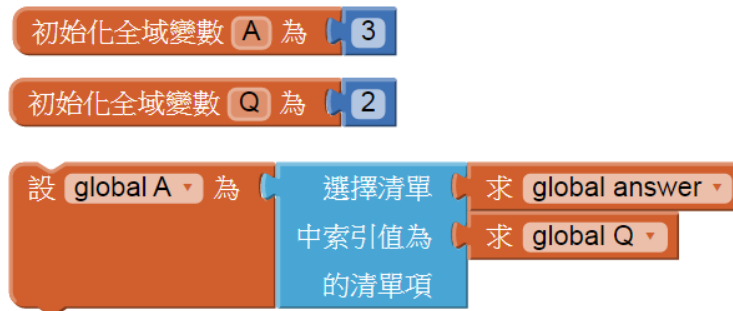
2. 一個清單名為"answer"，其內容如下：

索引	內容
1	100
2	80
3	250
4	70

(a) 在空格內填上適當的名稱和數值。



(b) 當執行以下的程式方塊時，變數 A 執行結果是甚麼？



(c) (i) 當執行以下的程式方塊時，變數 A 執行結果是甚麼？



(ii) 當執行以下的程式方塊時，變數 A 執行結果是甚麼？



自主學習 – 編寫筆記

1. 清單變數

一個清單變數能夠 _____



上方的程式方塊定義一個名為 _____ 的清單變數。它包含了 _____



上方的第一行程式方塊的功能是 _____

第二行程式方塊的功能是 _____

其他重點筆記

第五課

猜時間遊戲

5.1 計時器 (clock) 元件

有些應用程式需要使用到日期、時間和執行某個事件所需的時間，以下是一些常見的例子：

- 顯示現在的時間
- 記錄用戶完成一個任務所需的時間
- 控制一件物件在一個遊戲中顯示及消失

在這些情況下，我們需要在應用程式內加入一個計時器。在 **App Inventor**，計時器是一個「不可見元件」(**invisible object**)，因此，當我們在畫面編排介面拖曳一個計時器至畫面時，它不會顯示在屏幕上，它只會顯示在畫面下方的外圍。然而，我們在程式設計介面仍可以程式方塊操控這計時器和讀取它的時間數據。

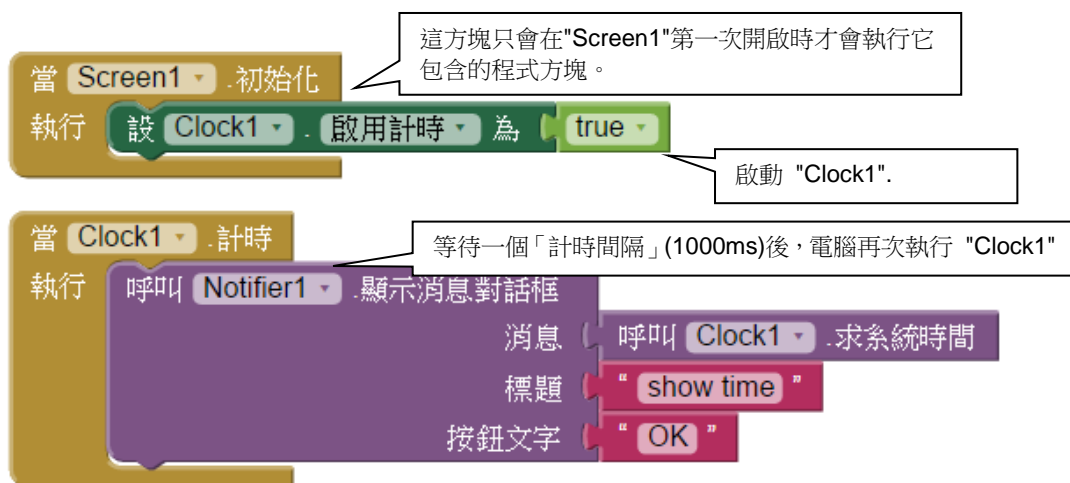


App Inventor 的計時器操作方法如下：

- (1) 在畫面編排介面先設定計時器的「計時間隔」。
計時間隔是應用程式等待多少千分之一秒才再執行計時器的方塊一次。



在以下的例子，我們應用一個名為"**Clock1**" 的計時器在每一秒鐘展示一個對話框，展示現在的時間。



活動 5A 簡單的計時器

創作一個 App Inventor 應用程式，在"**Screen1**" 插入一個「對話框」(Notifier) 。再在「程式設計介面」加入上方的程式方塊，執行應用程式然後回答問題。

問題

1. 對話框 "**Notifier1**" 是不是一個可見的物件？你是怎樣判斷的？

2. "call Clock1.SystemTime" 的程式方塊展示一串長長的數字，這串長長的數字是代表甚麼？

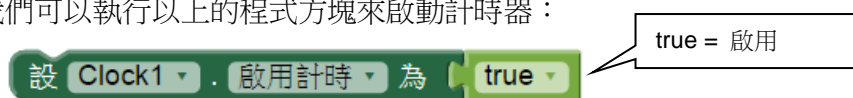
3. 如果我們想展示現在的日期和時間，我們該怎樣改變這些程式方塊？

備註：

有時我們不想程式一開始執行時便即啟動計時器，我們可以執行以下程式方塊「停用」計時器：



當需要計時功能時，我們可以執行以上的程式方塊來啟動計時器：

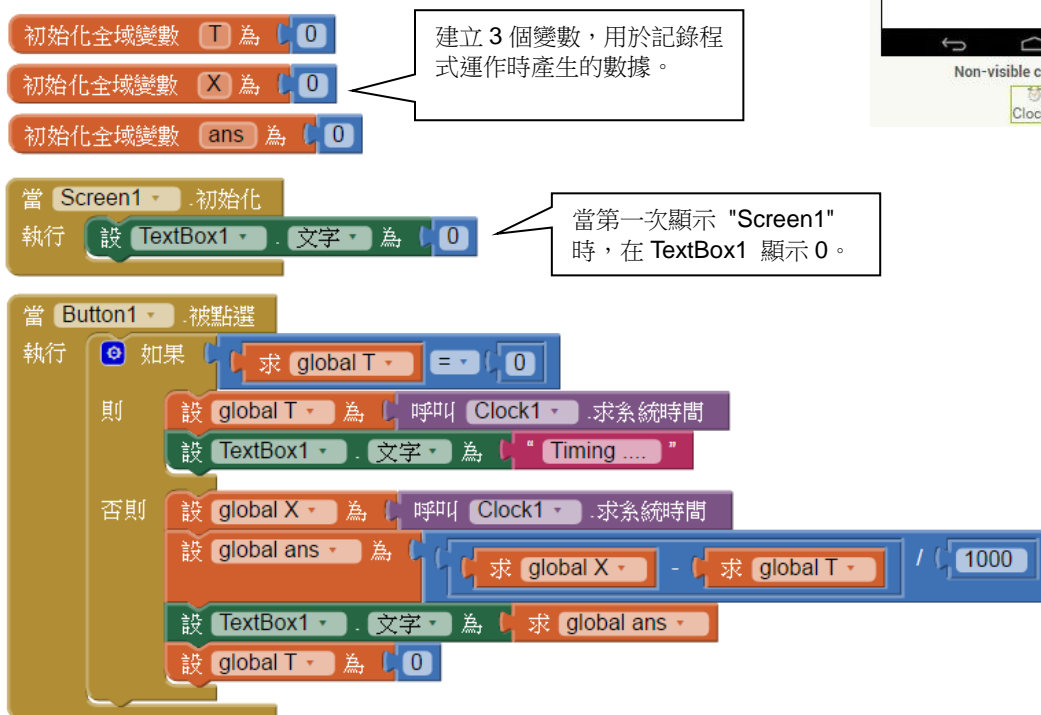


活動 5B 製作秒錶

我們在此活動製作一個簡單的秒錶，當使用者按下 [Start/Stop] 按鈕時，它開始計時，再按一下[Start/Stop] 按鈕後，秒錶會顯示兩次按下按鈕之間的時間。

請按照以下的指引建立秒錶應用程式。

1. 在 App Inventor 建立一個名為 "Simple Timer" 的專案。
2. 在 "Screen1" 插入右圖所示的元件。
3. 在「程式設計」介面插入以下的程式方塊。



4. 儲存專案，然後在模擬器執行你的程式。

問答題

1. 在空格上寫出各程式方塊的主要作用。

當 Button1 被點選
 執行 如果 求 global T = 0
 則 設 global T 為 呼叫 Clock1 求系統時間
 設 TextBox1 文字 為 "Timing"
 否則 設 global X 為 呼叫 Clock1 求系統時間
 設 global ans 為 求 global X - 求 global T / 1000
 設 TextBox1 文字 為 求 global ans
 設 global T 為 0

如果還未開始計時 (T=0) 則

否則

活動 5C 優化秒錶

優化活動 5B 的秒錶，加入以下的功能：

1. 加入一個 **RESET** 按鈕，它的功能是重設 **TextBox1** 的值為 0 (並且停止計時器元件操作)。
2. 每約 0.1 秒更新時間顯示一次。

在下方的程式方塊填上適當的內容，以完成上述的要求。



初始化全域變數 **T** 為 0

初始化全域變數 **X** 為 0

初始化全域變數 **ans** 為 0

當 **Screen1** 初始化

執行 設 **Clock1** 啟用計時 為 false
 設 **TextBox1** 文字 為 0

當 **Button1** 被點選

執行 如果 反相 **Clock1** 啟用計時
 則 設 **Clock1** 為
 設 global **T** 為 呼叫 **Clock1** 求系統時間
 設
 否則 設 **Clock1** 為

當 **Clock1** 計時

執行 設 global **X** 為 呼叫
 設
 求 global **X** - 求 global **T** / 1000
 設 求

當 **Button2** 被點選

執行 設
 設

延展活動 5 (反轉教室支援活動)

[中三] → [App Inventor 第 4 課]

修改活動 5C 的程式使它能以分和秒形式表示時間。(參考右圖)
列印程式方塊並把它貼在下列空格內。



自主學習 5

在 App Inventor 設計一個時計遊戲，遊戲玩法如下：

- 當玩家按下 "New Game" 按鈕，程式首先產生一個 2 至 9 的整數，儲存至變數 T，然後在屏幕顯示該數值。
- 當玩家按下 "Start" 按鈕，遊戲程式記錄現時的系統時間至變數 X。
- 當玩家按下 "Stop" 按鈕，遊戲程式記錄現時的系統時間至變數 Y。
- 如果 $(X-Y)$ 和 T 的值小於 1，玩家便勝出，否則輸掉遊戲。遊戲結果以對話框(notifier)展示。



反轉教室的短片展示了部份程式編碼，請在 App Inventor 複製該程式方塊至一新專案，並加入適當的程式方塊以完成整個遊戲程式。

列印程式方塊並把它貼在下開兩頁的空格內。

[illegible]

專題習作

動作遊戲：地球保衛戰

6.1 畫布 (Canvas) 和圖片精靈 (ImageSprite) 元件

動作遊戲(action game) 涉及很多移動的圖像。當這些物件互相碰撞時，它們會出現多種不同的結果，例如：

- 向相反方向移動 (反彈)
- 在屏幕上消失 (遊戲角色被摧毀)
- 更變角色的外觀 (顯示另一張圖像)

我們必須使用圖片精靈元件(在繪圖動畫類別)來創做動態和互動的物件，而不是只能顯示靜態圖像的圖片元件(在使用者介面類別)。同時，我們需要使用畫布(在繪圖動畫類別)設定圖片精靈可以在螢幕裡移動的範圍。畫布的設定包括畫布高度、寬度和背景圖片等。

在這專題習作，我們會建立一個射擊外星人的動作遊戲，玩家需要駕駛着飛船，射出火球擊退外星人。被擊中外星人會即時消失。當所有外星人被清除，玩家便勝出。然而，如果在遊戲期間，火球跌入下方的地球，玩家便輸了遊戲。為增加遊戲的難度，畫面中間加入三塊不會被破壞的隕石。火球擊中隕石後會反彈。以下是這遊戲的畫面設計。

The screenshot shows the OK_Proj_1 development environment. The central workspace displays a game scene with a spaceship at the bottom, several alien sprites at the top, and three stone sprites in the middle. A green 'START' button is at the bottom. The left sidebar shows the '元件面板' (Component Panel) with 'Canvas', 'ImageSprite', and 'Sprite' components. The right sidebar shows the '元件清單' (Component List) with a list of components including 'Canvas1', 'Ball1', 'alien3', 'alien4', 'alien5', 'alien2', 'alien1', 'spaceship', 'stone1', 'stone2', 'stone3', 'Button1', 'Notifier1', and 'Sound1'. The bottom right shows the '應用說明' (Application Description) panel with fields for 'AppName' (break_brick), '背景顏色' (Background Color), '背景圖片' (Background Image), '畫面關閉動畫' (Screen Close Animation), '圖示' (Icon), '畫面開啟動畫' (Screen Open Animation), '畫面方向' (Screen Orientation), '允許捲動' (Allow Scrolling), '狀態欄顯示' (Status Bar Display), and 'Sizing' (Fixed).

1 加入一個畫布作為背景、一個球形精靈和 9 個圖片精靈。

2 加入一個按鈕、一個對話框和一個音效元件。

3 把學校下載網址的圖像上載至素材部份。

4 設定 Canvas1 大小為 300X400 和背景圖像為 pic2.png

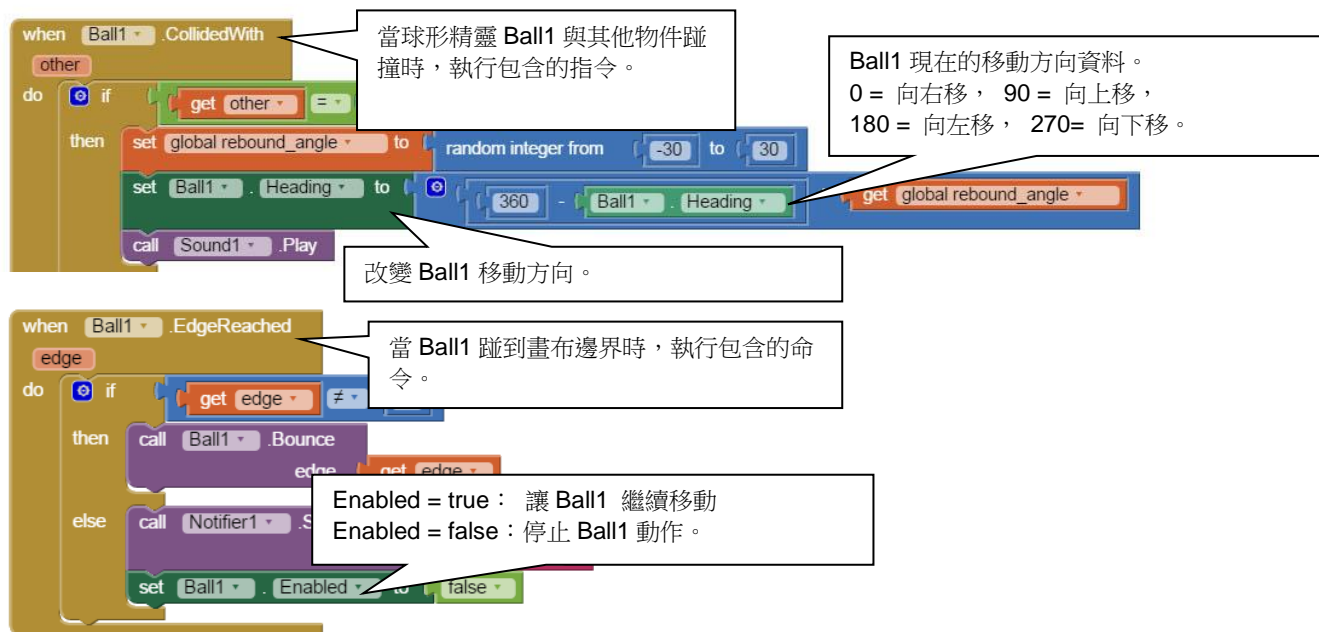
5 更改圖片的名稱

6 Specify the right PNG images to all imagesprites properly

插入 START 按鈕

6.2 使用球形精靈 (Ball)

App Inventor 提供了一個名為的「球形精靈」(ball) 的元件，它是一個可以在畫布(canvas)上走動的物元。它有後多控制指令讓我們運用它來建立動作遊戲。以下這專題習作所使用的球形精靈所具備的特別功能。



6.2 編寫程式概念

一個動作遊戲程式包括兩個部份：

(1) 初始化變數(Initialize variables)

設定遊戲開始時的分數、物件位置和管理遊戲運作的變數的數值。

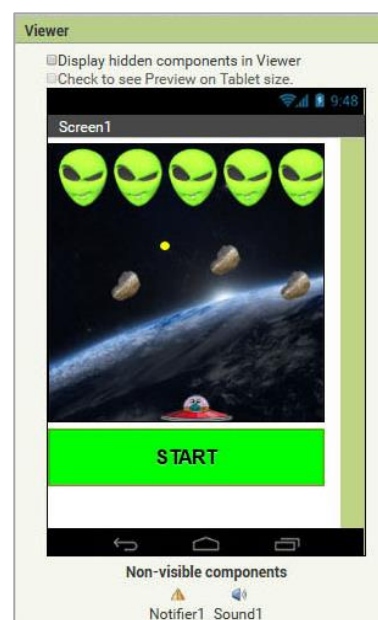
(2) 定義遊戲裡所有會發生的事件(event)的回應

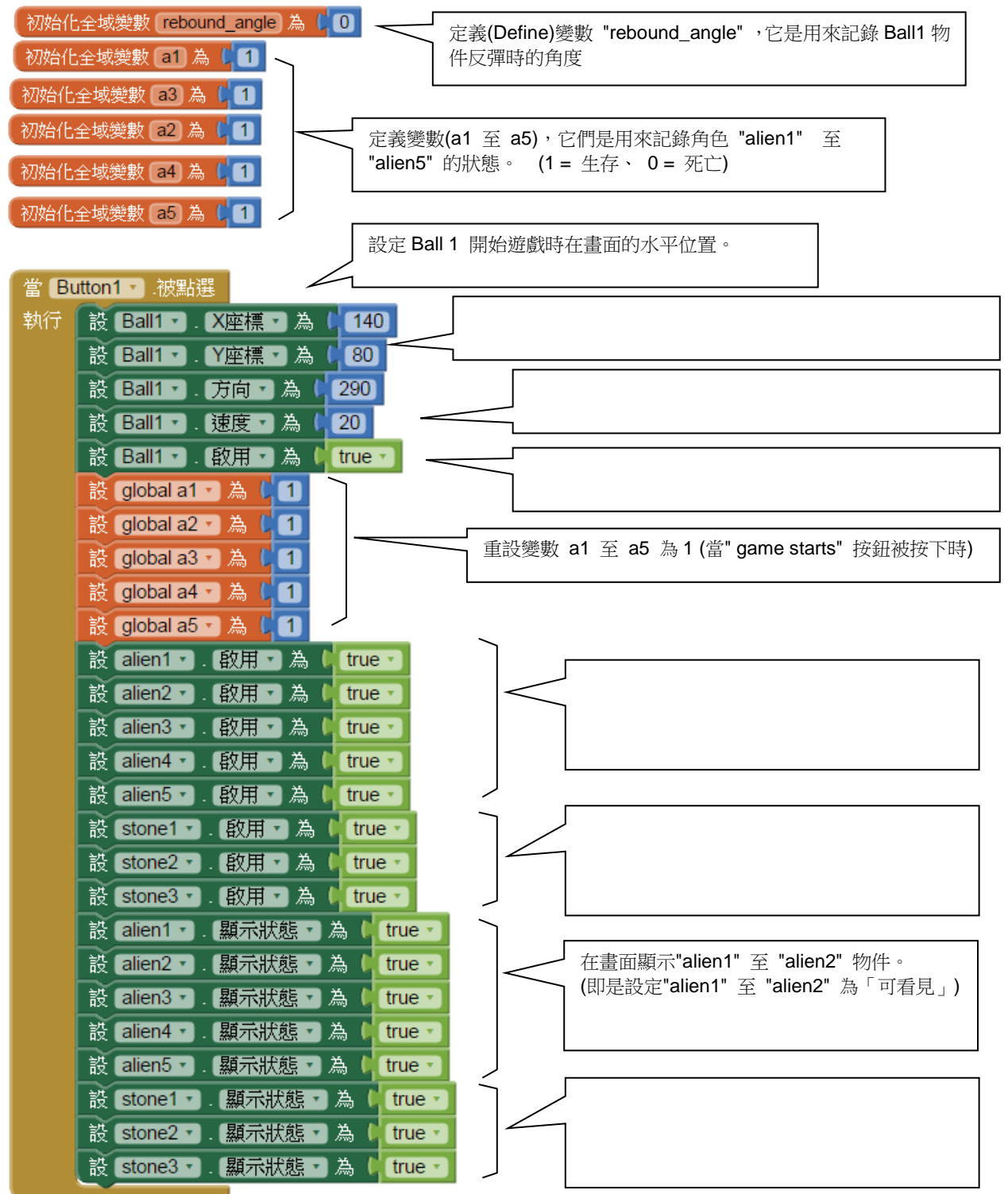
編寫程式方塊處理遊戲裡的物件對事件的回應，調整遊戲的計分，更新管理遊戲的變數的數值，以及決定遊戲是否完結。

活動 6A 編寫保衛地球遊戲

我們在這活動會在 App Inventor 編寫保衛地球遊戲的程式主體。

1. 建立一個新的專案，名為 "Defense_the_world"。
2. 根據第 28 頁的步驟建立遊戲的介面。
3. 轉到程式設計介面，參考第 30 頁，建立初始化遊戲的程式方塊。
4. 分析各程式方塊，在第 30 頁的方格簡單描述它們的功能。





5. 在所有程式方塊的空格上填上適當內容，以完成物件對不同事件的回應。

當 Ball1 到達邊界

執行

如果 求 邊緣數值 等於 -1

則 呼叫 Ball1 反彈 邊緣數值 求

否則 呼叫 Notifier1 顯示告警訊息 通知 "Game Over"

設 Ball1 為 false

如果 Ball1 碰到畫面的底部 (edge=-1) 則

如果不是碰到底部的邊緣 (edge ≠ -1)

當 spaceship 被拖動

起點X座標 起點Y座標 前點X座標 前點Y座標 目前X座標

執行 呼叫 spaceship 移動到指定位置

x座標 求

y座標 spaceship

如果 Ball1 碰到玩家的飛船，則 ...

當 Ball1 被碰撞

其他精靈

執行

如果 求 其他精靈 等於 spaceship

則 設 global rebound_angle 為 整數亂數從 -30 到 30

設 Ball1 方向 為 360 - Ball1 方向 + 求

呼叫 Sound1 播放

如果 求 其他精靈 等於 alien1

則 設 alien1 啟用 為 false

設 alien1 顯示狀態 為

設 global a1 為 0

如果 求 其他精靈 等於 alien2

則 設 alien2 啟用 為 false

設 alien2 顯示狀態 為 false

設 global a2 為

如果 求 其他精靈 等於 alien3

則

如果 求 其他精靈 等於 alien4

則

如果 求 其他精靈 等於 alien5

則

如果 求 其他精靈 等於 stone1 或者 求 其他精靈 等於 或者

則 設 global rebound_angle 為 整數亂數從 -30 到 30

設 Ball1 方向 為

呼叫 Sound1 播放

如果 求 global a1 + 求 global a2 + 求 global a3 + 求 + 求

則 呼叫 Notifier1 顯示消息對話框

消息 "You Win"

標題 "You Win"

按鈕文字 "OK"

設 Ball1 啟用 為

如果 Ball1 碰到外星人物件，則.....

如果 Ball1 碰到石頭物件，則 ...

列印你的程式方塊，並貼在下方的空白內。

延展活動

試在你的「保衛地球」遊戲加入以下一項功能，使它更有趣味。

1. 每當清除掉一個外星人，加快 **Ball1** 的移動速度 20%。
2. 在畫面加多一塊隕石。
3. 加入「遊戲難度」選項，難度愈高，玩家的飛船便更細小，火球的移動速度也愈高。
4. 任何額外的有趣遊戲設計。

列印你的遊戲畫面和程式方塊，夾附在本筆記內。