

使用 App Inventor 設計行動遊戲軟體

林致宇

摘要

智慧型行動裝置的普及，使得愈來愈多的行動創新應用服務被提出來。其中一個應用是使用於無所不在學習，學生們只要手持行動裝置便可在任何時間、任何地點進行學習，而為了引發學生學習的動機與興趣，學習內容必須具有趣味性，通常可搭配影音多媒體或遊戲來進行學習，然而這也使得學習內容的製作變得複雜，要一個非資訊專長的教師製作一個行動裝置上的教學遊戲軟體幾乎是不可能的。Google 針對 Android 作業平台的行動裝置，於 2010 年提出了 App Inventor 這套開發工具，它的主要訴求是希望非資訊專長的一般使用者也能開發自己專屬的手機應用程式，本論文便是探討利用 App Inventor 開發遊戲教學軟體的可能性，我們會以九格拼圖遊戲為例，說明如何利用 App Inventor 來開發遊戲。

關鍵字：數位遊戲設計、行動遊戲設計、App Inventor

Developing Mobile Games Using App Inventor

Chih-Yu Lin

Department of Applied Informatics and Multimedia, Asia University, Taiwan
lincyu@asia.edu.tw

ABSTRACT

With the popularization of smartphones, many creative mobile applications have been proposed. Some mobile applications are developed for ubiquitous learning so that students can learn at anywhere, anytime with their mobile devices. In order to enhance learning motivation, the learning content should be interesting. Usually, we can use multimedia content and games to enhance learning motivation. However, this makes producing learning materials complex and difficult. It is impossible for a teacher whose major is not computer science to develop learning games. In 2010, Google has announced a development tool called App Inventor that allows users to create applications on mobile devices adopting the Android operating system even though the users are not professional programmers. In this paper, we investigate the possibility of developing learning games by using App Inventor. We will use a jigsaw puzzle game as an example to present how to develop a mobile game by using App Inventor.

Key Words: Digital Games Design, Mobile Games Design, App Inventor

一、 緒論

隨著智慧型行動裝置的普及，使得愈來愈多的行動創新應用服務被提出來。其中一個應用是使用於無所不在學習，學生們只要手持行動裝置便可在任何時間、任何地點進行學習，這種學習模式常被使用於戶外生態教學[1, 7, 11]或博物館的導覽[5, 6]。而為了引發學生學習的動機與興趣，學習內容必須具有趣味性，通常可搭配影音多媒體或遊戲來進行學習[12, 13]，然而這也使得學習內容的製作變得複雜，要一個非資訊專長的教師製作一個行動裝置上的教學遊戲軟體幾乎是不可能的。

目前智慧型行動裝置上的主要作業平台，有 Apple™ 的 iOS™ 平台[10]與 Google™ 的 Android™ 平台[3]，在這些平台已經有不少遊戲開發工具讓開發者能夠輕易地開發行動裝置上的遊戲，例如 Unity[14], GameSalad[9], Corona[8]等，我們會在第二節對這些開發工具做更詳細的介紹。在 2010 年，Google 針對 Android 作業平台的行動裝置提出了 App Inventor 這套開發工具[4]，它的主要訴求是希望非資訊專長的一般使用者也能開發自己專屬的手機應用程式，由於使用 App Inventor 的使用者不需要具備高深的程式語言技巧，因此其非常適合拿來做為程式語言的入門教學工具，此外，對於非資訊專長的國中小老師來說，也非常適合拿來做為行動教學軟體的開發工具。

本篇論文將探討利用 App Inventor 來開發行動教學遊戲軟體的可能性，我們將以一個益智遊戲「九格拼圖」為例，說明利用 App Inventor 來開發行動遊戲軟體的流程，我們的目標是希望能讓想開發行動遊戲軟體的教師，能夠藉此範例，更容易地瞭解利用 App Inventor 來開發遊戲軟體的過程，進而設計出更多具教育意義的行動遊戲軟體。本論文的章節組織，首先在第二節我們會對一些現有的遊戲開發工具做一些簡單的介紹與討論，在第三節則對 App Inventor 做介紹，在第四節中我們則利用九格拼圖為例，詳細解說該遊戲的開發流程，最後我們在第五節對本篇論文做一個簡短的結論。

二、 文獻探討

行動裝置的普及也在教育領域開啟了一種新的可能性，產生了所謂的行動學習(m-learning)，行動學習在結合了情境感知的概念後，更產生出所謂的無所不在學習(u-learning)，關於行動學習與無所不在學習已經有不少的相關研究發表於期刊或會議上[1, 7, 11, 12, 13, 15]，其中也包含了討論如將遊戲結合至教學上的論文，限於篇幅有限的關係，我們無法做太詳細的介紹，讀者可自行搜尋論文並閱讀。

由於本篇論文的重點是討論行動遊戲軟體的設計，我們將針對數個已發表的遊戲開發工具做介紹。UNITY[14]是一個功能強大的遊戲開發工具，能夠設計常見的 2D 與 3D 遊戲，且目前已經支援 iOS 與 Android，然而若要使用進階的功能，是需要付費的。GameSalad[9]也是一套遊戲開發工具，其強調使用者不需寫任何的程式，因此很適合非資訊專長的開發者，然而目前 GameSalad 只能在特定的機器與作業平台上執行。Corona[8]是一套能夠於手機上開發應用程式與遊戲的工具，能支援 iOS 與 Android，它是免費的軟體，然而開發者必須具備一些程式語言的技巧。本論文所使用的 App Inventor，是 Google 於 2010 年針對 Android 平台提出的開發工具，能夠用來開發一般的 Android 應用程式，其強調開發者可以不是專業的程式設計師，且是免費的工具，因此很適合非資訊專長的開發者，而本論文將探討利用 App Inventor 來開發遊戲的可能性。

三、 App Inventor 簡介

App Inventor 是 Google 針對 Android 作業平台[3]的行動裝置，於 2010 年提出的以

瀏覽器為基礎的開發工具，開發者只要使用平常用來瀏覽網頁的瀏覽器便能開發 Android 裝置上的應用程式，而開發者所設計的應用程式專案(Project)會放在雲端儲存起來。圖 1 是一個範例，我們進入開發頁面後，就可以管理自己儲存於雲端的應用程式專案，使用者可以下載自己的專案儲存於自己的電腦上，並分享給其他人，其他人只要將所收到的專案，利用自己於 App Inventor 的帳號，再上傳到自己的空間即可。

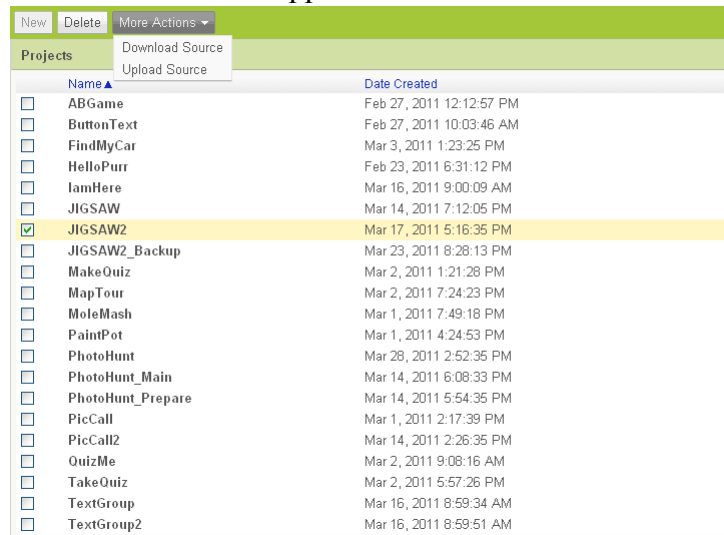


圖 1: App Inventor 專案管理畫面

App Inventor 所提供的開發工具可分成三類：介面設計工具(Designer)、程式行為編輯器(Blocks Editor)與模擬測試工具(Emulator)，下面我們將分別一一做介紹。

3.1 介面設計工具(Designer)

「介面設計工具」是用來將應用程式所需要用到的元件，如按鈕、顯示文字用的標籤等，擺放到手機螢幕上，也就是做使用者介面的設計，使用者於圖 1 的專案管理頁點選某一個專案或這建立一個新專案後就會進入此介面設計工具，如圖 2 所示：



圖 2: App Inventor 之介面設計工具

介面設計工具可再細分成五個部份：元件調色盤(Palette)、畫面預覽(Viewer)、使用元件清單(Components)、上傳媒體檔案(Media)與屬性設定(Properties)，下面將分別做介紹：

- 元件調色盤(Palette):
元件調色盤裡包含了許多內建的元件，如按鈕、顯示文字用的標籤、安排子元件擺放位置的元件等，元件可進一步分成「可看見」與「無法看見」，「可看見」的元件如按鈕，使用者可於手機的螢幕上直接看到按鈕，而「無法看見」的元件則如資料庫(TinyDB)、時鐘(Clock)、位置感測器(LocationSensor)等，我們無法於此一一介紹每個元件，然而讀者可於官方網站[4]查詢到各個元件的用途與使用方法。
- 畫面預覽(Viewer):
畫面預覽是讓開發者可以預先看見當應用程式完成後，手機的顯示效果，開發者可藉此畫面調整元件的擺放位置。
- 使用元件清單(Components):
當開發者從元件調色盤拉了元件到畫面預覽的視窗內時，該元件就會出現在使用元件清單裡，代表開發者使用了該元件，我們可以幫元件取一個有意義的名字，例如一個遊戲內可能有一個「開始按鈕」，玩家按下後便可開使進行遊戲，則我們可將此按鈕命名成 StartButton，如此可增加程式的易讀性。
- 上傳媒體檔案(Media):
有時我們會希望我們的應用程式內有一些影音多媒體，透過「上傳媒體檔案」的功能，我們可以將儲存在自己電腦內的影音檔上傳至 App Inventor 的伺服器。
- 屬性設定(Properties) :
當我們在使用元件清單裡點選一個元件後，該元件會反白，並且我們可以在屬性設定的視窗設定該元件的屬性，以顯示文字用的 Label 元件為例，我們可以在屬性設定視窗設定文字的字體大小、字體顏色、文字內容等屬性，當然稍後我們也可以在程式中修該元件屬性。

3.2 程式行為編輯器(Blocks Editor)

在「介面設計工具」畫面的右上方，我們可看到一個名為「Open the Blocks Editor」的按鈕，按下後就可以進入程式行為編輯器，如圖 3 所示：

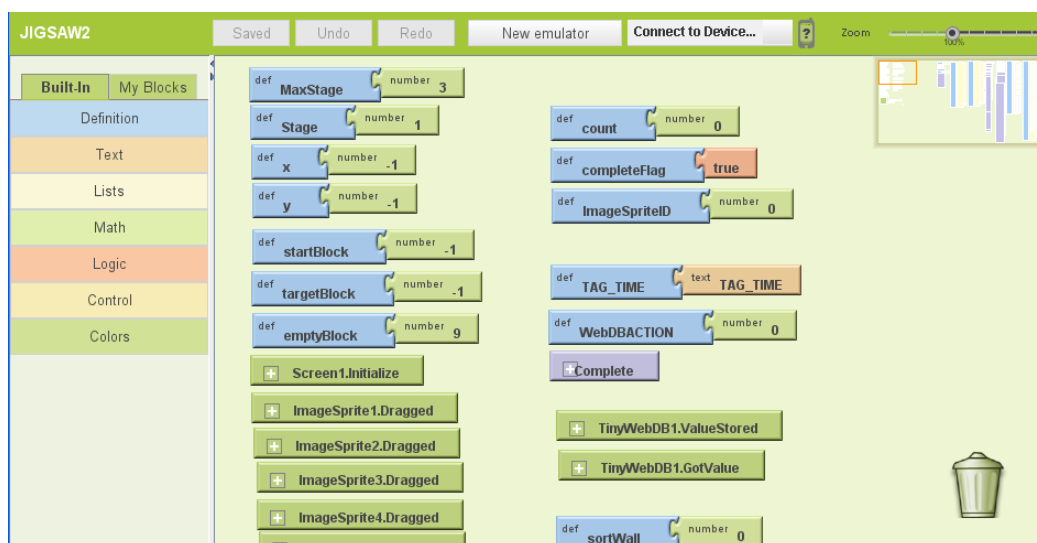


圖 3: App Inventor 之程式行為編輯器

先前已經提過，在「介面設計工具」中我們可以拉一些元件到手機畫面中，例如我們可以為我們的應用程式增加一個按鈕，然而這個按鈕目前沒有任何的作用，按下去不會有任何的反應，為了要讓這個按鈕發揮作用，我們必須在「程式行為編輯器」定義這個按鈕的行為，因此程式行為編輯器就是讓開發者定義程式行為，也就是寫程式的地方。

在「程式行為編輯器」的畫面左側有一些程式「積木」(Blocks)，透過這些積木的組合，我們便可以完成程式的行為定義，這些積木分為兩類：「內建積木」與「使用者自創積木」。內建積木提供了一些一般程式語言裡該有的功能，如變數的定義、流程控制(*if, foreach, while* 等)、簡單的數學函式等；使用者自創積木則是使用者自行定義的積木，我們於介面設計工具所新增的元件都算是我們自行定義的積木，而我們定義的變數也都是自行定義的積木。關於更多「程式行為編輯器」的說明，讀者可至 App Inventor 的官方網站[4]查詢。

3.3 模擬測試工具(Emulator)

當我們將介面設計好，並將程式行為定義好後，我們可以說已經完成應用程式的製作，此時我們會嘗試執行我們所設計好的應用程式，以便檢查程式是否有問題。如果我們有 Android 平台的手機，我們可在「程式行為編輯器」右上方的「Connect to Device」選項，選擇連線到我們的 Android 平台手機，如此便能測試我們的程式。然而如果我們沒有 Android 平台的手機，我們可以先按下「New emulator」，便可使用如圖 4 所示的模擬器，我們可於模擬器上執行我們的程式，用以檢查程式是否有問題。若程式執行有問題，我們就必須回到「介面設計工具」或「程式行為編輯器」將問題做修正後再重新測試。



圖 4: App Inventor 之模擬器

四、 九格拼圖遊戲之設計與實作

數位遊戲的總類相當多，如益智遊戲、格鬥遊戲、射擊遊戲等，由於我們是針對數位學習，希望能設計出搭配教學的遊戲軟體，因此本論文選擇了一個常見的益智遊戲：九格拼圖。九格拼圖的玩法說明如下：九格拼圖是將遊戲空間分成九格，其中的八格會放上圖片，只留下一個空白格，這些圖片是打亂放上的，遊戲的目的就要將打亂的圖片依正確的順序排好，為了達成這個目的，玩家可以將空白格旁邊的圖片移至空白格，而被移動的圖片其所在的格子就成為新的空白格，一直重覆這個動作，最後就可將圖片依正確的順序排好。圖 5 是一個範例，圖 5(a)是遊戲初始的畫面，畫面的右下角是空白格，一開始玩家可將空白格上方或左側的拼圖移至空白格，如此反覆將拼圖移至空白格，最後可將拼圖依正確順序排好，如圖 5(b)所示。

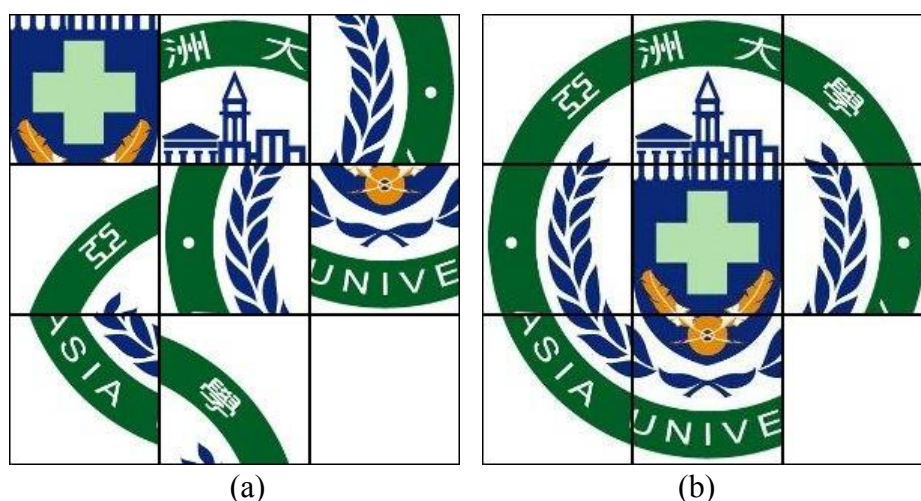


圖 5：九格拼圖範例

藉著這個九格拼圖遊戲，我們將說明利用 App Inventor 設計遊戲的流程，我們的目標是讓想製作遊戲的使用者，只要遵循此流程便能完成遊戲的開發，因此我們將在文中分享製作過程的心得，供遊戲製作者參考。遊戲的製作流程可分成四部份：功能設計、外觀設計、行為設計與遊戲的執行與測試，下面我們將分別說明。

4.1 功能設計

利用 App Inventor 設計遊戲的第一個步驟是決定遊戲所應具備的功能設計，以九格拼圖為例，除了能讓使用者移動拼圖這個基本功能之外，我們可能還希望有下述的功能：

- 計時功能：記錄玩家完成拼圖所花費的時間。
- 計步功能：記錄玩家移動拼圖的次數。
- 重新開始：讓玩家能讓遊戲重新開始，當然遊戲時間與所花步數會歸零重計。
- 選擇關卡：遊戲應具備複數個關卡供玩家選擇，以免產生厭倦感。
- 排行榜：讓使用者能夠上傳成績，並顯示出所有玩家的成績。

我們製作的經驗是，開發者可先完成遊戲的基本功能，其它的附屬功能可之後再添加，因此開發者可先跑完流程的四個步驟，完成一個雛型後，再回到此步驟，思考是否要添加額外功能。

4.2 外觀設計

第二個步驟是外觀設計，我們建議可依各個功能來設計所需要的外觀，下面將分別說明：

- 遊戲基本功能：
當我們使用 App Inventor 來設計含有動畫或移動圖片的遊戲時，通常我們會需要用到兩個元件：Canvas 與 ImageSprite。Canvas 可以想成是一個二維的空白畫布，這個畫布允許我們在其上畫圖或者讓我們放上可被移動的圖像，此外當使用者於 Canvas 上做觸控的動作時，App Inventor 提供了相對應的處理程序 (Procedure) 讓開發者來處理觸控時應該執行的行為。ImageSprite 則是可被移動的圖片，如同 Sprite 這個字所代表的意義，圖片能像小精靈般在 Canvas 內自由移動。在九格拼圖遊戲中，我們可以在 Canvas 內放入八個 ImageSprite，每一個 ImageSprite 代表一塊拼圖，如此我們便能讓使用者移動拼圖以進行遊戲。
- 計時功能：
計時功能是要顯示遊戲經過的時間給玩家，因此只要利用基本元件 Label 便可以完成計時功能的外觀。然而光是顯示文字用的 Label 是不夠的，先前已經提過 App Inventor 所提供的元件中可分為「可看見」與「無法看見」，此處我們需要一個屬於「無法看見」的元件 Clock，用以記算遊戲經過時間。經驗不足的開發者有時可能會忽略屬於「無法看見」的元件，開發者應把常用的屬於「無法看見」的元件記住，並隨時提醒自己是否需要用到那些元件。
- 計步功能：
計步功能是要顯示玩家移動拼圖的次數，因此只要利用顯示文字用的基本元件 Label 便可以完成計步功能的外觀。
- 重新開始功能：
重新開始功能是要讓使用者能重新開始遊戲，由於是希望玩家點選後才重新開始，因此須設計成按鈕的型式，亦即利用基本元件 Button 便可以完成重新開始功能的外觀。
- 選擇關卡：
選擇關卡功能是要讓使用者能選擇不同的關卡，也是設計成按鈕的型式，利用基本元件 Button 便可以完成外觀設計。
- 排行榜功能：
排行榜功能是要讓使用者能觀看網路上所有玩家的排名，直覺上也是設計成按鈕的型式，但是我們可以想見為了達成上述的功能，我們的畫面上已經有畫布、文字區塊、按鈕元件，已經沒有太多的空間讓我們顯示排名，此外排名是一個列表 (List)，因此我們應考慮使用 ListPicker 這個元件，這個元件的外觀跟按鈕類似，使用者按下後，會跳出另一個螢幕並將列表內容顯示出來。除了 ListPicker 之外，我們還需要一個屬於「無法看見」的元件 TinyWebDB，此元件是一個網路資料庫，此資料庫有兩個特性：(1) 所有使用相同應用程式的使用者皆能存取此資料庫，因此能達到資料的共享；(2) 開發者所設計的不同應用程式也能互享資料。第一個特性代表我們能讓遊戲的玩家上傳他們的成績並與大家共享，第二個特性則代表我們可以幫遊戲管理者另外設計一個應用程式來管理資料庫內容，不過這並不是這篇論文所要討論的。

經過上述的說明，我們便可以開始利用 App Inventor 提供的 Designer 將外觀實作出來，圖 6 是實作的結果，圖 6(a)顯示了利用 Designer 預覽的結果，圖 6(b)則顯示了圖 6(a)中使用到的元件名稱，為了讓元件能夠以水平的方式做排列，我們還利用了 App Inventor 提供的 HorizontalArrangement 元件。此外要提醒注意的是，我們還有兩個無法看見的元件：Clock 與 TinyWebDB。

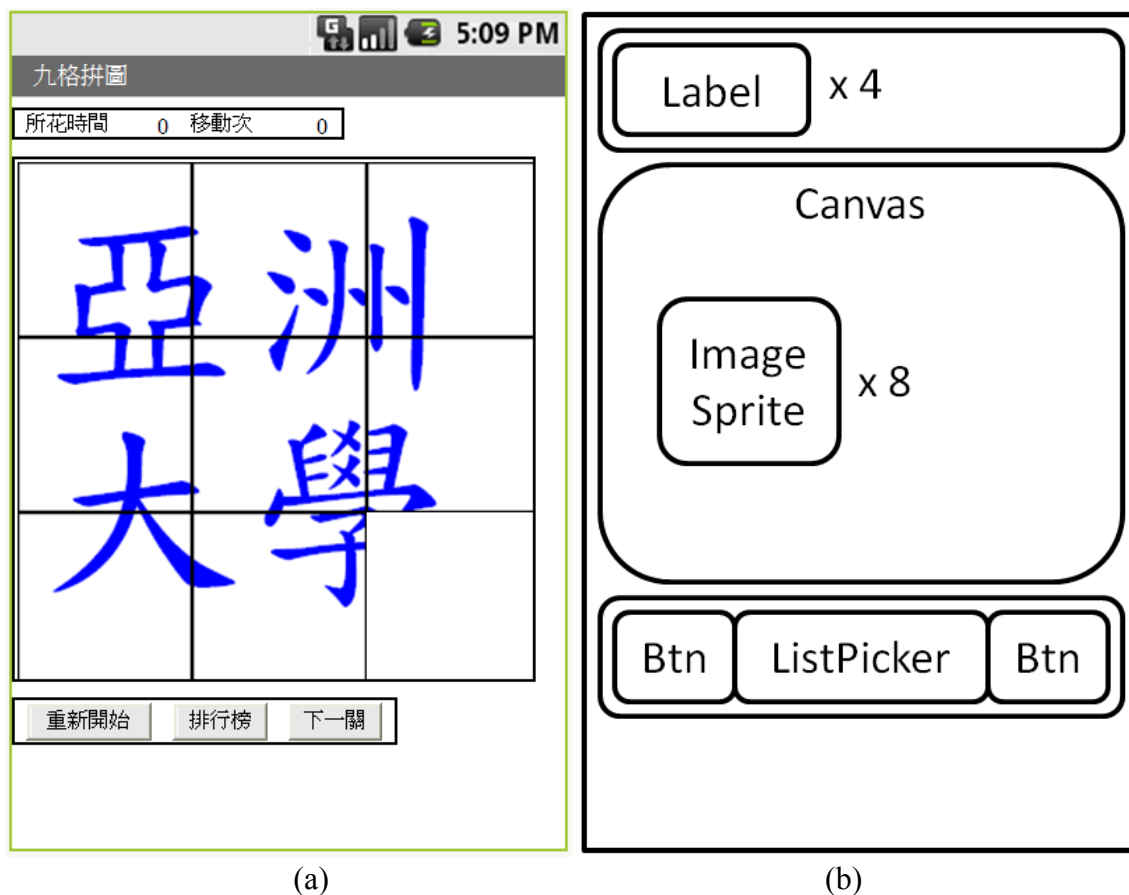


圖 6：九格拼圖外觀設計

最後要分享一個經驗給讀者，在設計外觀時，我們很難一次到位，例如當玩家完成拼圖時，我們可能需要利用能顯示文字的 Label 元件，或能發出文字訊息的 Notifier 元件，來告知使用者其已完成此關卡，這些元件都可以在事後回到 Designer 裡補上，也就是說雖然我們將開發流程分成四個步驟，使用者可隨時回到外觀設計的步驟增加必要的元件。

4.3 行為設計

在上一節中我們有設計了一些按鈕出來，如果使用者經過外觀設計步驟後，就直接嘗試去執行程式，會發現按鈕按下去不會有任何的反應，要讓按鈕或其它元件發揮作用，我們必須定義這些元件遇到某些事件（例如「按下」事件）時該有的行為，此即所謂的行為設計。我們一樣可依各個功能分別設計其行為，下面將分別說明：

➤ 遊戲基本功能：

遊戲基本功能要處理的就是拼圖的移動，先前已經提過，每一個拼圖是一個 ImageSprite 元件，App Inventor 允許我們定義此元件能被使用者碰觸(Touch)及拖曳(Drag)後該有的行為，在拖曳功能中，App Inventor 會告訴我們使用者從 Canvas 的哪個座標(二維座標)移動到哪個座標，但是不會告訴我們使用者移動的方向(亦

即上、下、左、右)，因此我們必須自行處理方向的判定，我們所設計的判斷方向演算法及部份程式碼如圖 7 所示，其中第 17 至第 27 行的區塊是考量到使用者在拖曳時不見得是完全的垂直或水平的情況而設計的：

```

1:  $currentX \leftarrow$  目前手指所在的 X 座標
2:  $currentY \leftarrow$  目前手指所在的 Y 座標
3:  $startX \leftarrow$  開始拖曳的啟始 X 座標
4:  $startY \leftarrow$  開始拖曳的啟始 Y 座標
5:  $diffX \leftarrow currentX - startX$ 
6:  $diffY \leftarrow currentY - startY$ 
7: if  $diffX = 0$  and  $diffY = 0$ 
8:     方向  $\leftarrow$  沒移動
9: else if  $diffX = 0$  and  $diffY < 0$ 
10:    方向  $\leftarrow$  上
11: else if  $diffX = 0$  and  $diffY > 0$ 
12:    方向  $\leftarrow$  下
13: else if  $diffY = 0$  and  $diffX < 0$ 
14:    方向  $\leftarrow$  左
15: else if  $diffY = 0$  and  $diffX > 0$ 
16:    方向  $\leftarrow$  右
17: else {
18:     $ratio \leftarrow |diffX| / |diffY|$ 
19:    if  $ratio > 3$  and  $diffX < 0$ 
20:        方向  $\leftarrow$  左
21:    else if  $ratio > 3$  and  $diffX > 0$ 
22:        方向  $\leftarrow$  右
23:    else if  $ratio < 1/3$  and  $diffY < 0$ 
24:        方向  $\leftarrow$  上
25:    else if  $ratio < 1/3$  and  $diffY > 0$ 
26:        方向  $\leftarrow$  下
27: }

```

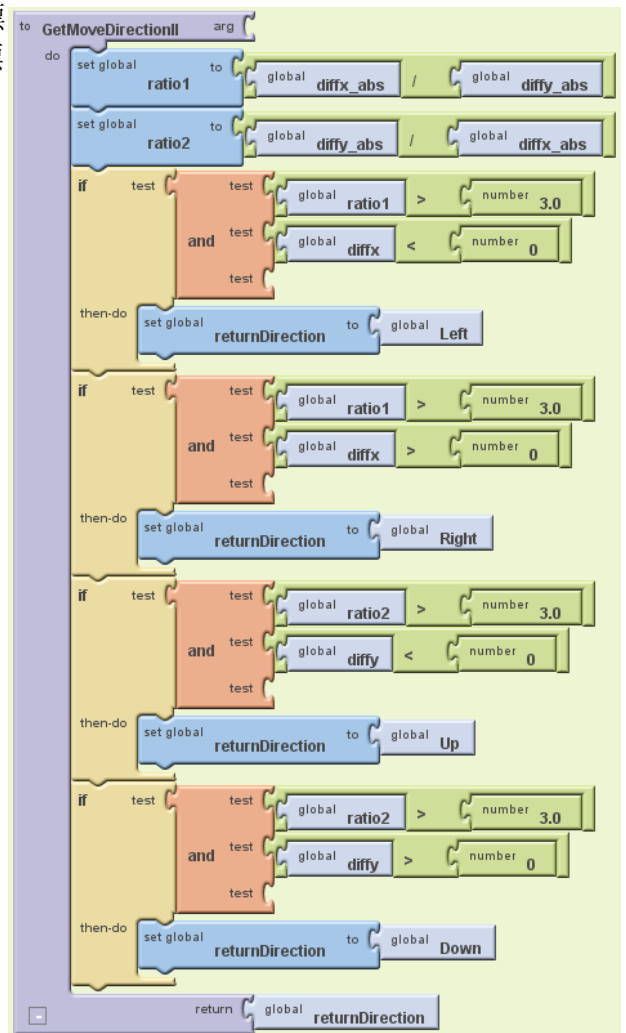


圖 7：判斷拖曳方向的演算法及部份相對應的程式碼

判斷完拖曳方向後，我們還必須判斷移後的方向是不是空白格，是空白格才能移動過去，否則就不該移動。而移動完成後，還要去判斷玩家是否已經完成此關卡，進而顯示出完成關卡的訊息，由於篇幅的關係，我們不在此將程式碼顯示出來，讀者可至我們的網站抓取相關的程式碼[2]。

➤ 計時功能：

計時功能的設計較為簡單，只要改寫計時器的 Timer 這個程序(Procedure)，當計時器設定的時間間隔到達時，這個程序就會被呼叫，我們只需要做兩件事，第一件事是把遊戲時間加一，第二件事是修改顯示遊戲時間的 Label 的內容，程式碼如圖 8 所示：

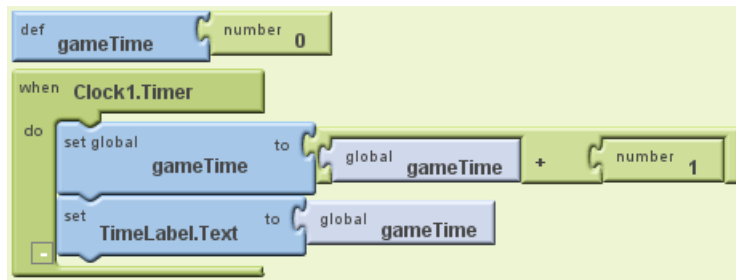


圖 8: 計時功能程式碼

當然遊戲重新開始或換新關卡時，遊戲時間要記得重設成零。

➤ 計步功能:

計步功能是寫在前面提到的「遊戲基本功能」裡，首先要定義一個變數記錄移動的次數，當玩家成功移動一塊拼圖時，我們就將此變數的值加一，並將變數值顯示於 Label 上。

➤ 重新開始功能:

由於「重新開始」是一個按鈕，我們只要改寫按鈕的 Click 動作即可，如圖 9 所示，由於使用者選擇下一關時也要重新初始化，因此我們將關卡的初始化寫成了一個名為 StageInit 的程序(Procedure)，「重新開始」和「下一關」這兩個按鈕都會呼叫此程序，此程序會執行一些關卡初始化的動作，例如時間與步數的歸零等，詳細的細節可下載我們所提供的程式碼[2]。



圖 9: 「重新開始」按鈕之程式碼

➤ 選擇關卡:

我們的遊戲程式中提供了一個「下一關」按鈕，一樣只要改寫按鈕的 Click 動作即可，如圖 10 所示，如前面所述，我們會呼叫 StageInit 做關卡的初始化，然而再此之前，我們還必須設定新的關卡編號，並更換圖片(每一個關卡的圖片皆和其它關卡的圖片不同)。

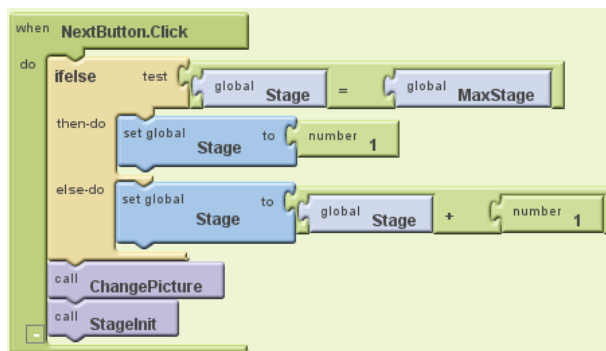


圖 10: 「下一關」按鈕之程式碼

➤ 排行榜功能:

此處我們只是為了示範 App Inventor 的網路資料庫功能，藉著網路資料庫我們可以讓全世界使用我們應用程式的使用者共享資料，在我們的程式中我們會將使用者完成關卡所花的時間列出。我們會在使用者完成關卡時，將使用者所花時間上

傳至網路資料庫，並抓取網路資料庫新的內容用以隨時顯示於列表(List)上。由於這部份的程式碼較為複雜，我們並不將程式碼列出，讀者可自行抓取閱讀。當然讀者可以做一些改進，例如讓玩家能輸入自己的暱稱，並於列表上改成呈現玩家的暱稱。

4.4 遊戲的執行與測試

程式設計完成後，最後就是利用模擬器來做測試，圖 11 顯示了我們所設計三個關卡。圖 12 則顯示完成關卡後的畫面以及排行榜畫面。完成測試後，整個遊戲的設計便完成了。

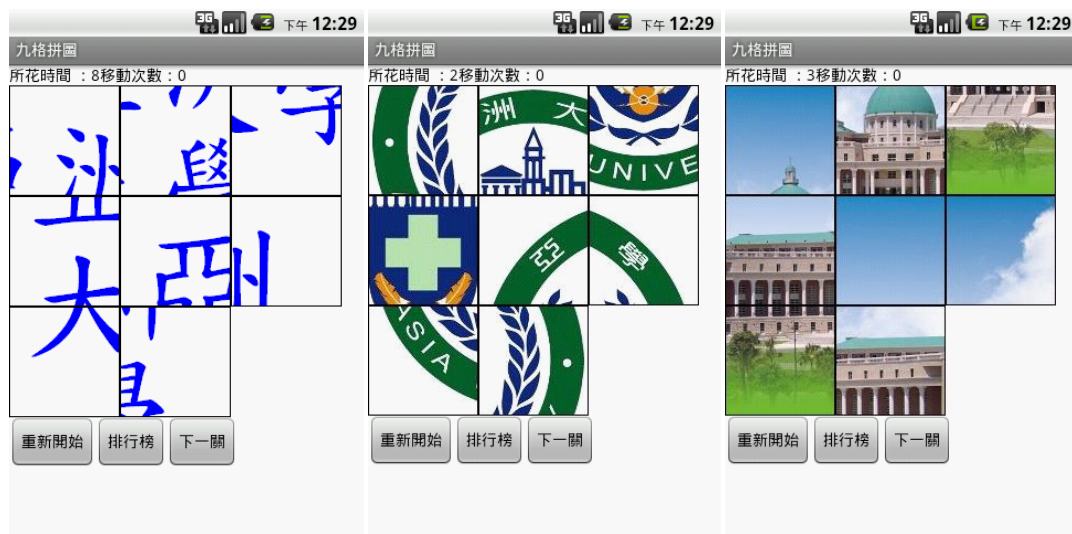


圖 11：遊戲的三個關卡

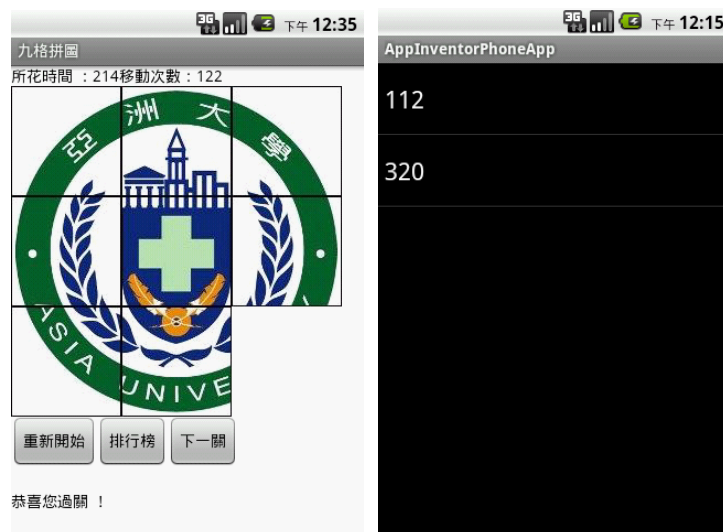


圖 12：完成關卡與排行榜畫面

五、 結論

本論文介紹了 App Inventor 這個由 Google 於 2010 年發表的開發工具，我們可以利用 App Inventor 做一些簡單的試題答問程式，App Inventor 的網站上也有提供相關的程式範例，然而學生對於測驗畢竟較不感興趣，因此我們嘗試利用 App Inventor 來開發

遊戲。在本論文中我們利用 App Inventor 設計了九格拼圖益智遊戲，這說明了我們的確有可能利用 App Inventor 來開發一個稍為有些複雜的遊戲，因此若要製作較為簡單的遊戲，如大家來找碴(Phone Hunt)，將更不是問題。我們相信未來會有更多利用 App Inventor 所開發的遊戲軟體被發表。

參考文獻

- [1] 蕭顯勝、馮瑞婷, "具情境感知式戶外生態教學系統之規劃與設計", 生活科技教育月刊, 39卷第5期, 2006年。
- [2] Android Programming Website, <http://asia.edu.tw/~lincyu/Android/android.html>
- [3] Android.com, <http://www.android.com/>
- [4] App Inventor for Android, <http://appinventor.googlelabs.com/about/>
- [5] Beat Fasel and Luc Van Gool, "Interactive museum guide: accurate retrieval of object descriptions," in Proceedings of the 4th international conference on Adaptive multimedia retrieval (AMR'06), Geneva, Switzerland, 2006, pp. 179-191.
- [6] Bruns, E., Brombach, B., Zeidler, T., and Bimber, O., "Enabling mobile phones to support large-scale museum guidance," IEEE Multimedia, Vol.14, No.2, 2007, pp.16-25.
- [7] Chu, H.-C., Hwang, G.-J., Tsai, C.-C., and Tseng, J.C.R., "A two-tier test approach to developing location-aware mobile learning systems for natural science courses," Computers & Education, Vol. 55, No. 4, December 2010, pp. 1618-1627.
- [8] Corona: The fastest and easiest way to create mobile apps, <http://www.anscamobile.com/>
- [9] GameSalad – Game creation for everyone™, <http://gamesalad.com/>
- [10] iOS Dev Center, <http://developer.apple.com/devcenter/ios/index.action>
- [11] Liu, T.-Y., Tan, T.-H., and Chu, Y.-L., "Outdoor natural science learning with an RFID-supported immersive ubiquitous learning environment," Educational Technology & Society, Vol. 12, No. 4, 2009, pp. 161–175.
- [12] Marchiori, E.J., Torrente, J., del Blanco, A., Martínez-Ortiz, I., and Fernández-Manjón, B., "Extending a game authoring tool for ubiquitous education," in Proceedings of the 3rd IEEE international conference on Ubi-media Computing (U-Media'10), Jinhua, China, 2010, pp.171-178.
- [13] Schwabe, G. and Göth, C., "Mobile learning with a mobile game: design and motivational effects," Journal of Computer Assisted Learning, Vol. 21, No.3, June 2005, pp. 204–216.
- [14] UNITY: Game Development Tool, <http://unity3d.com/unity/>
- [15] Zin, N.A.M., and Wong Seng Yue, "History educational games design," in Proceedings of the 2nd International Conference on Electrical Engineering and Informatics (ICEEI'09), Bangi, Malaysia, August 2009, pp. 269-275.