

# App inventor 導入工業設計背景學生程式學習案例

趙方麟

朝陽科技大學工業設計系

e-mail: flin@cyut.edu.tw

## 摘要

寫程式是在學習創意思考、有系統的推論、和團隊合作，這些技能不僅在各專業受用，更是生活中不可或缺的能力。設計背景學生過去少有學習程式，困難導致學習意願不高。為了提升互動設計的素養，導入 app inventor 於大三設計工學課程，以 9 周時間學習程式。發現非資訊背景學生學習程式需要不同教材與教學方法，並適度引用漸進式學習，減少初期障礙。此外在概念發想階段利用雛型工具，不至於限制創意範圍。在實現階段藉由跨領域配合，能提高創意完整度。設計背景者具有自身經驗才能知道技術的內涵，在與資訊背景學生合作中，修整概念設計範疇而使創意落實。

**關鍵詞：**設計背景、學習程式、App inventor。

## Abstract

Learning to write programs related to creative thinking, systematic reasoning, and teamwork. These skills not only utilize in the professional fields but also an indispensable ability. Design background students did not learning programs, sometimes students afraid of it. In order to improve the scope of interactive design, App Inventor was taught in the design engineering course. It arranged for 9 weeks. During programming learning, we found students need modified learning materials and teaching methods. Usually the progressive learning approach can reduce the initial obstacles. The use of virtual prototype tools in the concept stage helping extend creativity. In the implementation phase, cross-field cooperation can improve the integrity of the concept. Having own experience about technology, design students have chance to realize the concept of design by supporting of engineer.

**Keywords:** designer, programming, App inventor.

## 1. 前言

美國歐巴馬前總統曾提出投入 40 億美元資金到全美國學校，增強電腦科學教育，稱作「Computer Science for All」。部分直接撥給國小、國中、高中等學校，成立電腦科學部門。他表示：我們的經濟正在快速轉型，無論教育者或是商業領導者，都愈加認同電腦科學已經成為經濟機會與社會流動中的『新基本技能』[1,2]。電腦科學漸漸面向通識發展，美國有九成家長，希望孩子能學習電腦科學，但只有 1/4 的學校提供課程。

向下延伸後，在愛沙尼亞小學甚至一年級的學生都要學習撰寫程式。MIT Media Lab 團隊的計劃執行長 Mitch Resnick 展示學習網站[3,4]，讓孩子能輕鬆學會電腦程式並產生興趣。對於學習程式語言，Resnick 認為重點並非是要將所有年輕人培訓成程式設計師，而是透過學習開拓更寬廣的學習途徑。「當你學會閱讀，你便能藉著閱讀學習更多知識，程式設計也是一樣的道理；如果你會撰寫程式，你能透過程式語言學習到的事物將更為多樣。」互動已成為生活中的一部分，設計師自然不能除外。

### 1.1 國外研究

Xie(2016)分析 App Inventor 使用者的技能進展，MIT App Inventor 是一個非正式的在線學習環境，擁有 500 萬用戶和 1590 萬個應用程序。為了解人們如何使用與創建思維，以使用功能進展與思維概念的關係進行調查。隨機挑選用戶(至少已完成 20 個程序)，分析其能力的廣度和深度；使用者經由技能模組能構建更複雜的應用程序，強化開發能力；未來存在擴展至跨領域的非常見模式中。過去計算科學為主體的學習方式將會調整；工業設計跨領域的應用仍屬少見，這很可能出於對新技術的隔閡，值得發展。

Roberto(2016)探討南美洲的資訊教育，發現信息領域的技術專業人員短缺，移動開發工具的興起降低了進入障礙，有助於解決上述問題。

題。以短期應用程序開發營，提高中等教育學生選擇軟體工程的興趣。對參與研習營計的 60 名學生調查，發現對可能的職涯認識產生了積極的影響。顯示降低了進入障礙後，能鼓勵學生體驗並進入技術區塊，發現其中優美的特質，從而將自己的工作與之聯結。過去工業設計以造型為主體，在跨入互動設計領域時，因為沒有背景知識，導致成功的應用案例仍屬少見，形成跨入的隔閡，值得改善。

Sharefah (2016)探討沙烏地高中生資訊營之成效，AI2 已為 Android 手機創建功能齊全的應用程序，所以非程式師通過拖放代碼塊來建構程式，有效促進了初學者的學習。從學生創建的移動應用程序，顯示了研習營的效果，以及改進版的 AI2 亦增強了教學過程的順暢性。該研習營的作品以遊戲為主，很能引發初學者的學習動機。

Yerika(2015) 使用 App Inventor 結合歷史作業，以非洲裔非計算機科學學生為對象，推展 app 設計。所有學生都需要參加歷史課程，其中不存在背景差異。通過將 app 融入歷史課程來擴大少數族裔參與資訊學習，使歷史課程集成為有吸引力的展現。經由學生對技術的興趣，促進創建歷史應用程式。課程教學方法的基礎是建構主義學習，學生構建和生產知識，整合元件和技術；結果表明學生樂於參與，並熱衷於創建他們自己的應用程式。顯示跨領域的非常見模式，常能激發創意；這對工業設計學生亦具啟發性。

## 1.2 工業設計與互動設計

程式設計已經融入生活的各個層面，並且形塑未來生活與工作的想像。為了提升程式設計的基本素養，107 年課綱也將程式設計列入國中、高中階段必修課程。面對這項未來教育新趨勢，啟動不同的程式設計教育研習及校園推廣計畫值得期待。一小時程式設計是一個全球性的活動，已觸及數千萬個來自 180 多個國家的學生，讓任何人在任何地方都可以參與。其主要係透過視覺化[5]、積木式程式設計課程，讓非資訊專業背景的老師也能運用線上資源，引發學生對程式設計學習的興趣、了解運算思維、以解決問題為導向的學習。

設計背景學生過去少有學習程式，本校十年前曾設有選修課，於資訊概論內教授 Visual Basic；但是因為並不容易寫出能夠用於生活中的物件，學生意願不高。目前除了少數資訊互

動科系外，多數設計科系已無開設。本於在當代生活中，程式設計已經融入生活的各個層面，並且形塑未來生活的想像。為了提升設計的基本素養，雖然目前尚未列入課規，本系嚐試於 104 年導入 app inventor 於設計工學課程內，安排 9 周時間讓工業設計背景學生學習程式。

## 2.教學內容

非資訊背景學生學習程式需要不同教材與教學方法，工業設計系課程首先使用的教材為 App Inventor 2 中文版(圖 1)，作者：蔡宜坦、旗標出版 (2015 版次)，取用的內容主要包含下列實例：

第 01 週：App 程式開發，HelloPurr 展示

第 02 週：作業 2-9 Piano.aia 展示

第 03 週：作業 3-12 temp.aia 展示

第 04 週：作業 4-18 Paint\_2.aia 展示

第 05 週：作業 5-20 Ballot.aia 展示

第 06 週：作業 7-8 Movie.aia 展示/\*設計

第 07 週：作業 8-16 Book.aia 展示/\*試作

第 08 週：\*概念討論

第 09 週：\*概念發表與展示

進行中發現，在第五周之後部分學生無法跟上；探究其原因是非資訊背景學生在學習程式時，對於細節(例如避免輸入錯誤)尚不能掌握。當面對小障礙時，因為信心不足而逃避。因此調整教法，引用漸進式學習(圖 2,3)，減少初期障礙。此部分課程引入了 Prof. Wolber 著作[6]，與其第五章內容請參考；而漸漸能克服問題。漸進式學習是在同一個課題中，首先以基本功能展現，順利完成之後再漸漸加入一般化考量，與處理個別狀態問題(例如輸入超出範圍防止)。處理個別狀態問題往往涉及特定型態篩選，因為未能模組化，容易使學生進入細節而忽略全貌。

此外程式流程圖的導入，對導入工業設計背景學生程式學習也有幫助。因為學生習慣處理造型設計與視覺資訊；而學習寫程式在創意思考之外，仍需要有系統的抽象推論。視覺化的流程圖落實了思考步驟，易有助邏輯推演或與他人溝通。



圖 1 App Inventor 2 中文版

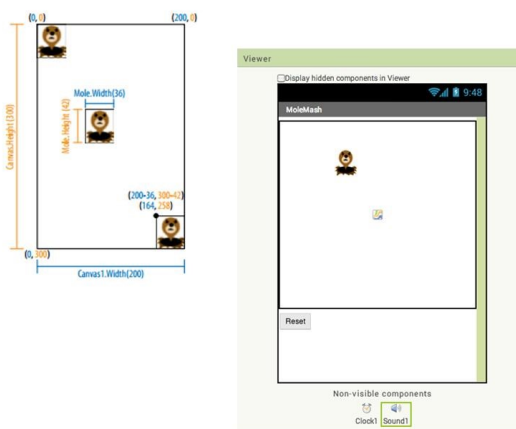


圖 2 Prof. Wolber [6] The MoleMash user interface

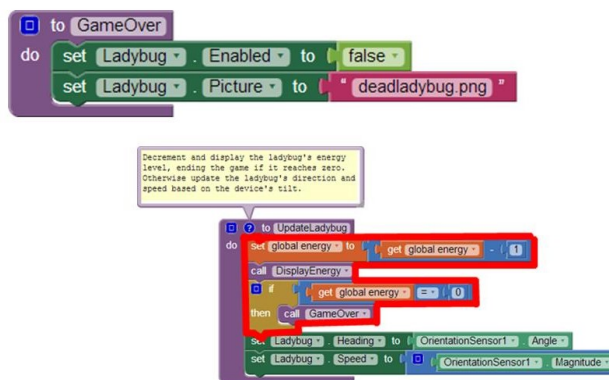


圖 3 漸進式細節導入[6]

第 08 週的概念討論是由學生提出欲解問題，並在第 09 週：概念發表時、展示於手機。非資訊背景學生著重互動介面的合宜性，因此我們在概念階段是利用 POP (paper prototype) 雛型工具，展現操作方式(圖 4)。先在紙上繪出想法，再加上設定的連結(link)，即可於 POP 平台操作；當可操作物件在手機展示後，才能體驗設計的合理性，不致於限制了創意範圍。

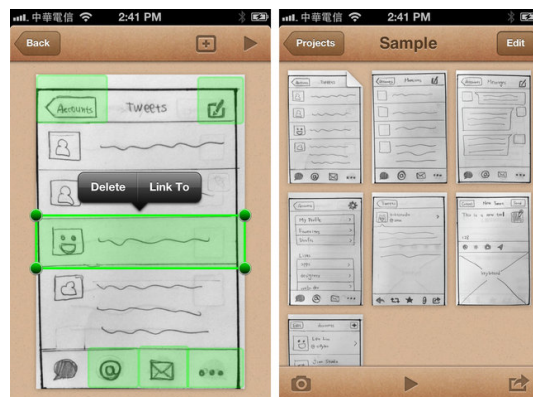


圖 4 Paper prototype 雛型工具[7]

由於時間限制，在實現階段無法分配足夠時間，只有部分完成程式區塊。除了寄望利用跨領域配合提高創意完整度之外，利用 POP 雛型工具時，學生想法過度擴張而無法實現。需要縮減過度擴張的概念，並且切割出最主要互動元素，優先以程式實現。

例如互動畫面(screen)數量增加時，互動畫面之間的參數互傳需要使用 Tiny DB 指令，對初階使用者構成困擾。經由過度擴張與適度縮減，此時因為非資訊背景學生已經有自身經驗，知道可行技術的內涵；當與資訊背景學生合作時，就能減少磨合時間，增進概念設計落實的機率。

### 3 創意設計例

創意教學首先需要了解周遭、在生活經驗中發現痛點，經由設計轉化形成亮點；再結合資訊技術的優勢，以程式實現連結需求端。學生發想過程會藉由創新法則，協助觀視周遭、脫離習慣領域在生活經驗中找尋獨特處，形成創新。

參與本課程學生共計 55 名，其中男性 30 位、女性 25 位。發想 APP 與應用為該科目的期中作業，發想提示從生活中尋找議題。期中作業只需做到提案階段，雖然經歷了 8 週的初步學習，為了避免學生程式能力限制了發想的範圍，因此並未強制使用程式表達。僅針對使用程式表達者個別加分。

經初步討論，消除部分常見構想之後；統計學生發想議題，其中以健康類別為最多、其次則為生活幫手。部分發想議題如下：

- 食譜、美食推薦、提醒取餐
- 生活計時、作業提醒

- 旅遊推薦、機車停車、換錢
- 拍照後尋物、穿著搭配、相片整理
- BMI、食物熱量、就醫提醒。

茲以五個提案為例子，呈現學生創意構想與思維方式。

### 3.1 牙套日記(陳品聿)

通常做這類矯正，都會在意別人眼光，擔心進展的程度。因此 APP 能紀錄牙套安裝的時間、歷程、變化，與醫生約診時間、費用等(圖 5)，透過朋友分享減少過程中的焦慮，協助找回自信。



圖 5 牙套日記圖示

### 3.2 寵物刺蝟(李易璉)

可以自行設定刺蝟的名字以及年齡，點進 Temperature 可以看到現在刺蝟生存環境的溫度。當溫度位於 24-27 度之外時，會顯示黃色無表情符號，表示溫度稍稍偏離；此時主人可以在遠端調整(圖 6)。

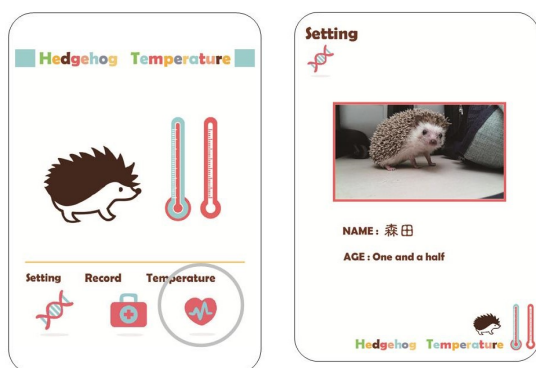


圖 6 寵物刺蝟設定頁面

### 3.3 心情、運勢(林鈞儒)

紀錄有關於自己穿搭的記錄 APP，今天我

的運勢是如何呢？我今天穿搭該搭配什麼顏色呢？才讓今天更美好呢？希望 APP 內建智識為你解答(圖 7)。



圖 7 心情運勢之程式塊

### 3.4 睡眠輔助(林欣嫻)

紀錄睡眠狀態(圖 8)，透過阿法波的樂音輔助改善睡眠。在頻域中，腦波最基本的分析是於頻譜上分為多個區帶，主要有  $\alpha$  波、 $\beta$  波、 $\gamma$  波、 $\theta$  波、 $\delta$  波。 $\alpha$  波，出現在深度睡眠。

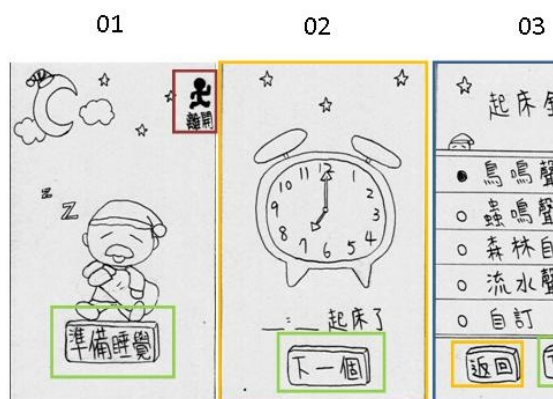


圖 8 睡眠輔助概念與程式

### 3.5 母子互動(江家紋)

Goodcare 是一個親子間的通訊軟體，家長可以隨時檢視孩子的位置，也可以跟孩子互動，透過此應用程式，讓家長再忙都能與孩子



互動。家長接收到孩子傳送的訊息，即可回傳；下方對話框出現“commihg soon honey”，顯示家長能接到孩子的時間(圖 9)。點選即可回到主選單。



圖 9 親子間的通訊

#### 4. 學生回應

課程並未要求參與學生填寫問卷，僅將部分學生心得回應摘錄，以了解半學期中的學習經驗與學生的內在變化。

- (1). APP Inventor 確實不難操作，且淺顯易懂，針對不同需求，你不用摸索太久，便有可能做出你想要的 APP。對於科技的時代，尤其近幾年，新一代設計展越來越走向科技，我認為這對於我們未來是一大利器，你可以將自己的設計創意，搭配獨特的科技元件，寫出屬於自己的程式。
- (2). Inventor 可以用最簡單的方式表達想到的互動形式，而且在創意與想法上面沒有限制讓人很好發揮。不論是小遊戲還是複雜的 APP 都能用它來製作，是一個非常有趣的程式工具。現在市場中許多好用的 APP 就是不經意發現問題後而設計出來的，在製作上也不需要花太多的時間，所以非常適合我們，在無聊時也可以隨意點點看說不定就做出了很好用的 APP。
- (3). 剛開始聽到要教 APP 程式以為很難，不過其實還蠻簡單的。但他是需要用一些邏輯去做設定，現在只是學個皮毛而已，個人感覺時間不足；是可以再深入學習的。雖然能力不充裕仍可以改善，未來能活用這個程式，在設計上會有很大的幫助。
- (4). 課程讓我了解原來製作一個 APP 其實很簡單，不用學很多看不懂的程式語言就能上手。在製作 app 的過程中，我也會去思考

使用者在使用時的習慣，或是各個指令之間的關聯都要先計畫好，才能形成一個有價值的應用程式，在未來製作畢業專題時也許能應用到這個資源。

- (5). 我第一次接觸到如何製作一個簡單的 app，在運算的邏輯上也可以做設定。就像一塊拼圖，要找到完成的形狀有很多種，但不一定最完美接合。APP Inventor 有它的步驟，但是也可以在裡面修改一些事情，讓它變得很不一樣。這也是我學習中體會到，有些因為軟體而致富的公司或人才，他們是怎麼思考設計與發明的。
- (6). 這次學到很多有趣的東西，一開始對 app inventor 很不熟悉，覺得模組之間的串連有迷惑，不太知道要放什麼。後來練習了幾次，就慢慢熟悉其間的關係，最後用手機掃 QR code 看到成品就覺得蠻酷的，以後有時間一定要好好專研耶！對於我們這樣的初學者，這是個做 app 的基礎，如果真的運用的很好，一定可以產生出不一樣的火花，也可以讓產品有另外的表現方式或是使用方式。另外的 pop 模擬工具我也覺得是個很棒的虛擬 app，我學到了很多新的東西！
- (7). 我們雖然只是學到一些皮毛，但還是從中發現許多有趣的程式設計，一個好的程式設計開發，是需要很久的時間去完成的，使用這個軟體，我們可以創造一個屬於我們自己的程式，很有挑戰性。

#### 5. 結語

非資訊背景學生學習程式需要不同教材與教學方法，尚有待建構。

1. 引用漸進式學習，能減少初期障礙。在概念階段利用雛型工具，不至於限制創意範圍。在實現階段須利用跨領域配合，才能提高創意完整度。
2. 非資訊背景學生有自身經驗，知道技術的內涵，才能與資訊背景學生合作。寫程式就是學習思考與有系統的推論、和團隊合作[8,9]，這些技能不僅在各專業領域都受用，也設計師不可或缺的能力。
3. 學習程式需要漸進式教材與適才適性教學方法。APP Inventor 具有維護機制，善用各種模組能使概念設計與發明落實[10]。中文版雖然方便閱讀，但是複雜模組中，時有因文字版本造成指令衝突或者

輸出停頓之困擾，正式開發時仍然建議以英文版本為優先。

## 參考文獻

- [1] [https://www.youtube.com/watch?v=AI\\_dayIQWV4&feature=youtu.be](https://www.youtube.com/watch?v=AI_dayIQWV4&feature=youtu.be)
- [2] <http://tedxtaipei.com/articles/10-places-where-anyone-can-learn-to-code/>
- [3] [http://ai2.appinventor.mit.edu/?locale=zh\\_TW](http://ai2.appinventor.mit.edu/?locale=zh_TW)
- [4] [https://sites.google.com/site/cs107f14/student\\_s](https://sites.google.com/site/cs107f14/student_s)
- [5] Hwang W Y, Wang C Y, Hwang G J, et al. A web-based programming learning environment to support cognitive development. *Interacting with Computers*, 20(6): pp.524-534. , 2008.
- [6] Wolber, David, et al. *App Inventor*. " O'Reilly Media, Inc.", 2011.
- [7] <https://play.google.com/store/apps/details?id=in.woomoo.pop&hl=zh>.
- [8] Spertus, Ellen, et al. "Novel approaches to CS 0 with app inventor for android." *Proceedings of the 41st ACM technical symposium on Computer science education*. ACM, 2010.
- [9] Gray J, Abelson H, Wolber D, et al. Teaching CS principles with app inventor//*Proceedings of the 50th Annual Southeast Regional Conference. ACM*, pp.405-406, 2012.
- [10] Gestwicki, Paul, and Khuloud Ahmad. "App inventor for Android with studio-based learning." *Journal of Computing Sciences in Colleges*, 27.1, pp.55-63, 2011.
- [11] X. Liu and H. Xu, "Reform on college fundamentals of computer course based on mobile computing," *11th International Conference on Computer Science & Education (ICCSE)*, Nagoya, pp. 890-893, 2016.
- [12] B. Xie and H. Abelson, "Skill progression in MIT app inventor," *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, Cambridge, pp. 213-217, 2016.
- [13] J. Espinoza-Guzmán, L. González, S. Forero and A. M. González, "Towards a methodological guide to share knowledge when creating open educational resources," *XI Latin American Conference on Learning Objects and Technology (LACLO)*, San Carlos, pp. 1-8, 2016.
- [14] F. Turbak, D. Wolber and P. Medlock-Walton, "The design of naming features in App Inventor 2," *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, Melbourne, pp. 129-132, 2014.
- [15] Y. Jimenez and C. Gardner-McCune, "Using App inventor & history as a gateway to engage African American students in computer science," *Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, Charlotte, NC, USA, pp. 1-2, 2015.
- [16] R. Munoz, T. Barcelos, R. Villarroel, C. Becerra, R. Noël and F. Ponce, "Using app inventor to change perceptions about software engineering," *35th International Conference of the Chilean Computer Science Society (SCCC)*, Valparaíso, pp. 1-9, 2016.