

從 Scratch 到 Raspberry Pi：中小學程式設計課程規畫

From Scratch to Raspberry Pi: Course Planning for Primary and Secondary Programming Education

高慧君

臺北市立南港高中

* koma@mail.nksh.tp.edu.tw

【摘要】 本文介紹了適用於國小、國中、高中三階段之程式設計課程規劃方式與教學工具。在國小階段，筆者建議使用 Scratch，以案例式教學引導學生創作電腦遊戲；教師並應將程式設計概念融入遊戲創作過程中，而不只是讓學生按圖施工。到了國中階段，則應將程式設計教學納入 STEM (Science, Technology, Engineering, Math) 教育之中，結合 Scratch 與 Arduino 進行機器人程式設計，而後再使用 Greenfoot 引導學生開始接觸文字式程式語言。在高中階段，筆者建議使用低價的硬體 Raspberry Pi 講授 Python 程式設計，除了讓學生進一步增進文字式程式設計能力之外，也能從中學習諸如作業系統、電腦網路等電腦科學相關知識。

【關鍵字】 中小學資訊教育、程式設計教學、STEM、Scratch, Greenfoot, Python, Raspberry Pi

***Abstract:** This article describes how to plan programming courses and select programming tools for elementary and secondary school students. The author suggests teaching game programming with Scratch in elementary classrooms and adopting project-based instruction. Instructors are also reminded to embed programming concepts when designing projects to prevent students from simply following recipe-style direction. Programming instruction at the junior-high school level should become part of the STEM (Science, Technology, Engineering, and Math) education. It can be implemented by letting students program robots with Scratch and Arduino, followed by text-based programming in Greenfoot. At the high school level, students can be taught Python programming using Raspberry Pi to further develop their text-based programming skills as well as learn computer science concepts such as knowledge about operating systems and computer networks.*

Keywords: K-12 education, Programming Instruction, STEM, Scratch, Greenfoot, Python, Raspberry Pi

1. 前言

筆者擔任高中電腦教師即將屆滿二十年。這些日子中，最想做的就是發展程式設計教材，一來因為在筆者的學習經驗中，高中自學 PE2, Lotus1-2-3 等應用軟體都能輕易上手，但想自學 Dbase 資料庫軟體時，卻不得其門而入。有幸筆者於大學時進入國立臺灣師範大學資訊教育學系就讀，於大一上、下學期分別學習 Pascal 和 C，到了大二修習資料結構、大三修習演算法。這些課程雖然不是很有趣，但大大增進了筆者的程式設計功力。大四的專題研究課程中，筆者選擇了多媒體製作專題，自學 Director，製作了系上的多媒體簡介與學校的校慶光碟。這四年的磨練培養了筆者的程式設計能力，也逐漸理解自己為何在高中時無法成功自學 Dbase。

筆者於大學畢業後輾轉至台北市南港高中任教。為使學生有較為愉悅的程式設計學習經驗，筆者嘗試了許多方法，例如使用了 ExcelVBA、JavaScript、Asp、Flash、VB，但都宣告失敗。在這些課程中，學生雖能馬上做出成品，卻缺乏程式設計的內涵。2010 年時，筆者得知國立臺灣師範大學林美娟教授編製了 Scratch 教材，至偏鄉地區教小學生程式設計，學生反

應良好；此外，林教授也向筆者推介了當年在 SIGCSE 研討會中引起廣泛注意的 Greenfoot 軟體。筆者就此開始接觸 Scratch 和 Greenfoot，並投入與其相關之教材與教具開發至今。

2. 各學習階段之課程規畫

2.1. 國小階段：Scratch（積木式程式語言）

Scratch（其程式設計環境如圖 1）是以拖曳、組裝積木的方式撰寫程式，而積木的組裝本身就隱含了語法檢查的功能，因此不必擔心會有語法錯誤的問題。其次，Scratch 的積木上可顯示中文指令，不至於帶來學習上的語言障礙；而官網上來自世界各國的許多作品，其積木式程式碼也都可以被翻譯成中文讓學生觀摩學習，因而創造了無國界的學習環境。最後，運用 Scratch 可輕易地完成各式各樣的互動遊戲。對於數位原生的新世代而言，他們雖然玩過許多電腦遊戲，卻並不瞭解遊戲背後運作的原理，因此，若能讓他們學習以 Scratch 開發電腦遊戲，應可獲得與玩遊戲截然不同的體驗。

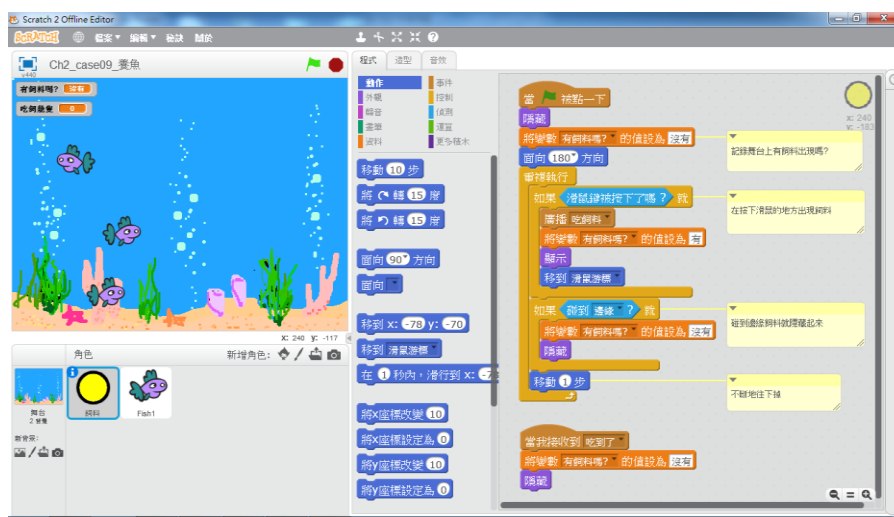


圖 1 Scratch 程式設計環境

2011 年時，筆者將一年以來針對 Scratch 的研究心得編寫成教案，發表於教育部高中資訊學科中心網站，並開始在自己的班上實施 Scratch 教學，同時將教案擴充編撰成授課講義。2013 年 5 月筆者進一步將授課講義加以擴充，出版了「程式設計邏輯訓練—使用 Scratch」一書。該書的精華在於程式篇的鋪陳是以螺旋式呈現程式設計概念與相對應的案例，亦即後面的案例不只出現新的概念，也會複習舊的概念。在教學過程中，為了確認學生不只是按圖施工，而確實能理解概念，筆者為每一個概念設計了一個小小的練習題。筆者另依實際教學進度調整教材，編寫成 Scratch 12 堂課，也為每一個案例和練習題增加了心智圖的整理，於 2015 年 4 月出版了「Scratch 2.x 用積木玩程式設計」國小用書。此書之主要特色如下：

(一) 經由遊戲創作來學習程式設計

有別於傳統程式設計教材以數學或商業應用的範例或練習題為主，本書採用互動遊戲創作來提昇學習興趣與成就感。遊戲本身提供了必須達成的具體目標，解法不再侷限於標準答案，學生亦可自由發揮創意擴充遊戲功能。

(二) 案例式學習

傳統程式設計教材多半以程式語言的主要概念為單位，依序編排教學單元，本書則是以案例來介紹程式設計的重要概念。案例的編排是以知識中心的角度，在每個案例中呈現一個重要的程式設計概念，讓學生可從實作案例的過程中理解概念，將概念運用於作品中，並進一步將其遷移至其它案例。隨著所學的概念愈多，學生所能完成的遊戲作品也愈來愈複雜。

Scratch 所標榜的特色是低地板（門檻低）、高天花板（作品複雜度高）、牆面寬廣（能容納不同興趣的學習者）。國內雖已有許多國小在電腦課中讓學生學習 Scratch 程式設計，但筆者認為目前小學生所達到的天花板並不够高。雖然這份 Scratch 12 堂課教材是根據筆者在高一實施四年的經驗匯集而成，但筆者認為小學生是有能力學會這 12 堂課的，因為學習程式設計並不像練啞鈴，年紀大、力氣大就比較佔優勢；它反而像是練馬步，所需要的是紮實的基本功。坊間的 Scratch 用書雖多，但大部份流於勞作性質，欠缺程式設計概念的鋪陳，僅只提供詳細步驟讓學生依樣畫葫蘆地完成作品，以致學生雖能完成作品，卻無法同時習得程式設計概念，遑論遷移至其它作品的創作上。筆者所編製的 Scratch 12 堂課（如下表），每個案例都呈現一個重要的程式設計概念。這個知識結構乃是根據筆者近十年來編寫程式設計教案的心得歸納而成。

Scratch 12 堂課內容

第 1 堂課 Scratch 基本介紹	第 7 堂課 雙向選擇結構：打地鼠
第 2 堂課 第一個動畫故事	第 8 堂課 全域變數：猴子吃香蕉
第 3 堂課 計次式迴圈：馬兒跑步	第 9 堂課 全域變數：打魔鬼
第 4 堂課 條件式迴圈：貓狗賽跑	第 10 堂課 角色變數：射蝙蝠
第 5 堂課 無窮迴圈：魚兒水中游	第 11 堂課 角色變數：養魚(上)
第 6 堂課 單向選擇結構：電流急急棒	第 12 堂課 角色變數：養魚(下)

2.2. 國中階段：應用 Scratch 於 Arduino 機器人程式設計

就程式設計教學而言，在國小階段若能確實培養學生的 Scratch 能力，則到了國中階段即可搭配 Arduino 開放硬體進行 STEM 教育。何謂 STEM 呢？STEM 就是科學（Science）、科技（Technology）、工程（Engineering）、與數學（Mathematics）的整合教育。

STEM 源自美國政府在 2006 年提出的《美國競爭力計畫》。該計畫強調美國的中小教育應培養具有 STEM 素養的人才，並稱其為全球競爭力的關鍵。美國總統歐巴馬並於 2011 年推出《美國創新戰略》時指出，美國未來的經濟增長和國際競爭力取決於創新能力，而為了推動創新教育，公部門和私部門應聯合加強 STEM 教育。2012 年歐巴馬總統發布《2012 年預算及中小學教育改革藍圖法案》，投入了超過 2 億元的預算推動 STEM 教育，預計兩年內招聘 1 萬名 STEM 教師，並於未來 10 年中培養 10 萬名 STEM 教師。美國國會又於 2015 年 10 月通過了《2015 STEM 教育法案》。由此可見，STEM 在美國不只是教改方向，更代表著美國對於國家競爭力的未雨綢繆與強烈企圖心。

南港高中於 103 學年度申請臺北市教育局國中部精進計畫，其目的在於發展以探究為基礎之「資訊」與「生活科技」統整教學，使其成為具校本特色之課程。本課程以 Scratch 軟體培養學生邏輯思考與問題解決能力，同時引導學生運用開放硬體 Arduino 電子元件及智高積木製作實體成品，使學生得以在研究與實作過程中，探討科學知識與技術在日常生活的應用，並培養工程實作能力。圖 2 所示為此一統整課程之課程架構。

機器人系統與課程架構分析

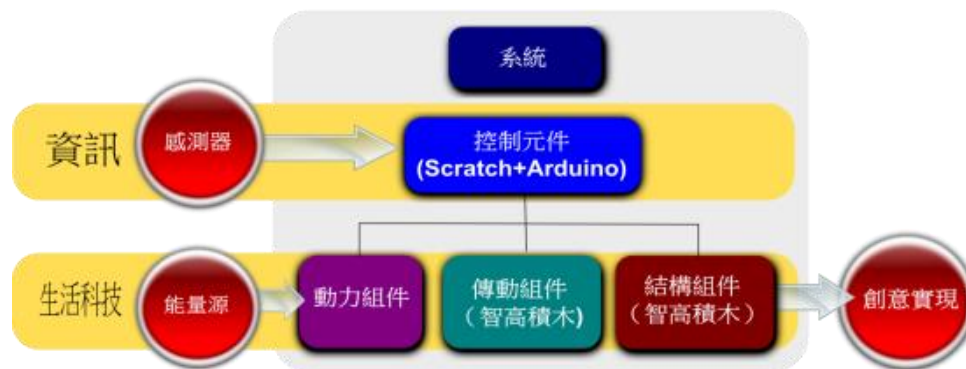


圖 2 課程架構圖

此統整課程之教材採用了賴鴻州老師編製之「智高 Scratch (S4A) 互動智能積木：動手玩創意 20 堂課」。103 學年度上學期先在聯課活動中讓 10 位國七、國八學生針對機器人程式設計進行「初體驗」(圖 3)；103 學年度下學期則於國七生活科技課程中讓學生以實體積木組裝機器人。課堂上每 3 人一組共用一套智高積木組，全班共分成 9 組，全年級各班共用 9 套智高積木組。在分組活動的過程中，我們著重培養學生的責任感與榮譽心，藉由分組競爭培養團隊合作能力。

國七學生在生活科技課程中組裝機器人之餘，另於資訊課中學習 Scratch 程式設計，並於 Scratch 課程結束之後，也進入 Arduino 課程，與生活科技課程同步進行上述 S4A 20 堂課。經過一學期的統整課程之後，大多數學生都能分組完成一部可以用電腦控制的機器車。

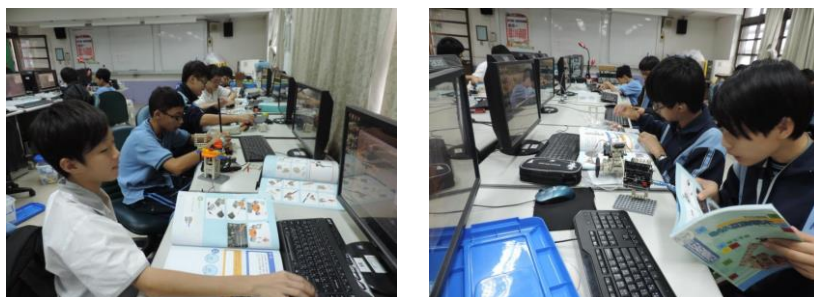


圖 3 國中部聯課活動學生專心組裝機器人



圖 4 利用段考下午辦理國八機器人競賽

104 學年度上學期國八生活科技課程實施了藍芽夾罐車及機關王課程，並於第一次段考的第二天下午舉辦機器人闖關大賽(圖 4)。各班先在每週 1 節的生活科技課堂上舉行班級競賽，從全班 9 組中選出優勝的兩組參加全年級競賽。競賽內容包含三關，第一關是「清除障礙—夾罐車」之設計與製作：學生使用智高積木組實作一輛具有夾物功能之遙控車，並撰寫

程式以控制機器車執行夾罐動作；第二關是「直線飆速—智高積木動力車」之設計與製作：學生使用智高積木組，運用齒輪、皮帶輪、或鏈輪等機械概念實作一部競速車；第三關是「龍爭虎鬥夾夾樂-夾罐車結構」之設計與製作：在第一關的設計之上，強化夾罐車的結構強度，使其能承受車輛碰撞干擾，並在規定時間內將不同顏色之瓶罐夾回自己陣地；各顏色瓶罐有一對應分數，以夾回瓶罐分數之總和決定名次。藉由這次的競賽，國七學生充分展現他們在資訊與生活科技整合課程的學習成果。圖 5 至圖 8 為學生完成的部分作品。



圖 5 學生創意作品

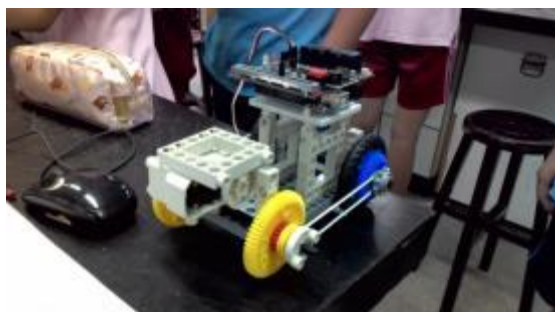


圖 6 學生創意作品



圖 7 機關王作品



圖 8 機關王作品

Arduino 機器人實作過程中，學生必須結合理化課程的電子電路知識、資訊課程的電腦程式設計、生活科技課程的自走車機構原理與馬達運轉方式等知識，將創意發想付諸實現。課程活潑有趣，更提供了高度的挑戰性，可說是很好的活化教學模式。

2.3. 國中階段：Greenfoot（文字式程式語言）

積木式語言雖然容易上手，可讓學生很快地產生成就感，但就像小孩子寫作文，因為認識的國字不夠多，只能寫注音文一樣，終究只是過渡期；當他們認識的國字夠多時，即應擺脫注音文的寫法，因為豐富的文字能讓他們表達更高層次及更多樣化的想法。同理，積木式程式語言也只是輔助表達的工具而已，若能由積木式程式語言提升至文字式語言，將可大大增進學生的程式設計表達能力。此時，Greenfoot 就是一個很好的選擇。

2011 年在筆者研究 Scratch 一年之後，又想起林美娟教授推介的 Greenfoot，於是開始在網路上搜尋相關資料。記得筆者剛大學畢業時，物件導向程式設計興起，筆者曾嘗試自學 Java，但發現大多數 Java 書籍都是前四章介紹語法，第五章開始才講解物件導向概念。語法本身相形容易，但第五章之後的物件導向概念非常抽象，令人難以理解如何用它來設計程式。這個困惑在筆者研究了 Greenfoot 之後才豁然開朗。那種感覺就像以前是在非母語的情況下學英語，只能死記一些單字與文法規則，而在接觸了 Greenfoot 之後，則像是在母語的環境中自然而然地學習英語。基於筆者透過 Greenfoot 自學物件導向程式設計的成功經驗，也就試圖將自己的成功經驗推廣到學生身上，於是在 101 學年度進行首次嘗試。筆者將學期的前 8 週用

於講授 Scratch，然後銜接 8 週的 Greenfoot，結果發現學生從積木式語言過渡至文字式語言有著很大的鴻溝，於是筆者決定暫停講授 Greenfoot。直到 104 年 10 月 Greenfoot 推出 3.0 版，強化了編輯器的許多功能，才又讓筆者燃起捲土重來的熱情。

舊版的 Greenfoot 編輯器直接讓學生寫 Java 程式，使得學生往往在分號、大括號等語法細節之間困擾很久。新版的 Greenfoot 3 (圖 9) 強調「以框架為基礎的編輯」(frame-based editing)，它將分號和大括號去除，並將每個程式區塊的程式碼自動標示與縮排；此外，使用者只須鍵入關鍵字首，即可叫出相對應之程式區塊框架，供使用者填寫細節。以選擇結構 if 為例，使用者只需鍵入 i，系統就會跳出 if (條件式) 敘述和它的程式區塊框架。從 Greenfoot 官方消息得知，他們為了這個新版的功能奮鬥了四年。筆者的感想是：Scratch 和 Greenfoot 就像是學術界研發的新藥，而中小學教師與其埋首在自己的小世界裡土法練鋼找解藥，還不如跟上世界潮流脈動，向外尋求最好的解藥。可惜教育界習於故步自封，加以教育問題多如牛毛，往往需要很多時間與努力，才能促成些許改變。

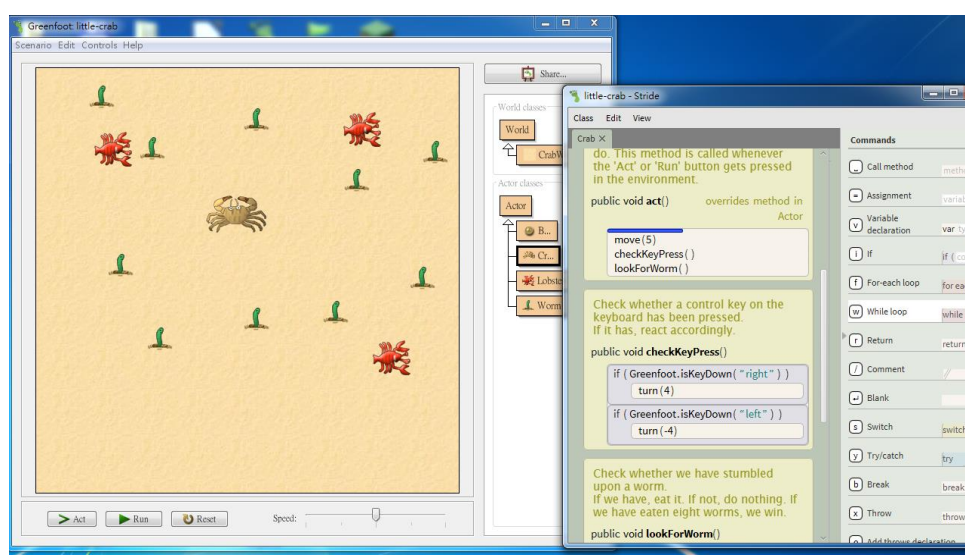


圖 9 Greenfoot 程式設計環境

2.4. 高中階段：Python 和 Raspberry Pi

學生在國中、小學了 Scratch 和 Greenfoot 之後，到了高中可學些什麼呢？筆者的選擇是 Python，理由如下。以閱讀能力的培養為例，我們在小學三年級之前通常先「學習閱讀」(“Learn to Read”)，有了閱讀技能之後即可「從閱讀中學習」(“Read to Learn”)。以此類推，筆者認為國中、小階段的程式設計教學重點在於讓學生「學習寫程式」(“Learn to Code”)，到了高中階段則需進一步「從寫程式中學習」(“Code to Learn”)。

如前文所述之 STEM 整合教育，在中小學義務教育階段實施程式設計教學的目的，不在於將所有學生培養成未來的軟體工程師，而在於將程式設計教學視為中小學科學與數學整合教育的一環。筆者在 2015 年 8 月參加教育部資訊教育總藍圖世界咖啡館活動時，得知台大物理系石明豐教授正在推廣 VPhysics。VPhysics 是甚麼？V 代表視覺化 (Visual)、虛擬化 (Virtual)，和透過電腦程式語言 (VPython)，最後邁向 Victory of Physics。有些物理現象即使透過文字描述、圖片表達、動畫展示，仍然無法幫助學生理解，若能透過程式設計模擬物理現象，將有助於學生認識我們身處的世界是如何運作的。筆者也因此頓悟，高中程式設計教學的目的，除了繼續充實學生的程式設計技能及增進電腦科學知識之外，亦應幫助學生

藉由程式設計學習物理、化學、數學知識。為達此目的，樹莓派（Raspberry Pi）（如圖 10）是一個很好的教學工具。

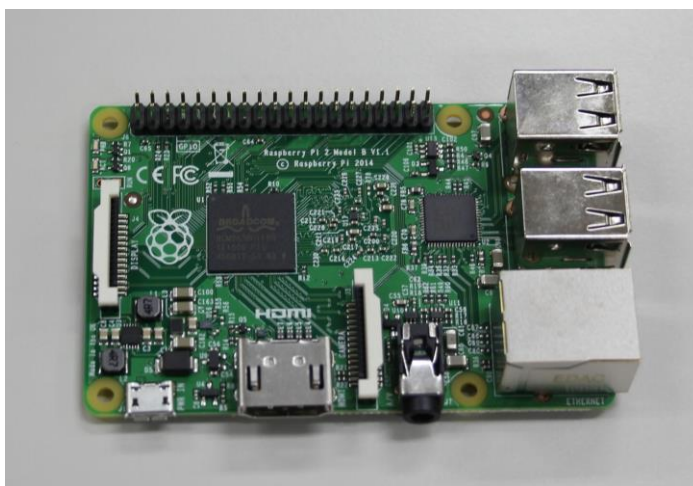


圖 10 樹莓派 Raspberry Pi

樹莓派(Raspberry Pi)是英國營利組織樹莓派基金會（Raspberry Pi Foundation）所開發的一片信用卡大小的電路板，搭載處理器（ARM CPU）和 Linux 系統，以 SD 卡為硬碟，目的是為學校的電腦科學教育提供低價的硬體和自由軟體。「樹莓派基金會」創辦人—艾本·厄普頓（Eben Upton）2006 年在劍橋大學任教期間，有感於每年申請電腦科學系的高中生越來越少，而且主修電腦科學學生的程式設計技能愈來愈不如人意，他因此期許自己能開發一項滿足下列四項條件的設備：

1. 必須能支援程式設計，而且盡可能地支援多種程式語言。
2. 必須能支援遊戲和影音，讓孩子們覺得有趣。
3. 必須夠精巧、夠堅固，讓孩子們可以隨身帶到學校去
4. 必須夠便宜（價格定在 25 美元，因為這和一本教科書的價錢差不多）。

2008 年時，厄普頓離開劍橋大學，開始為博通公司（Broadcom）工作，而博通所開發的晶片正適合開發上述設備，因此厄普頓和 5 位合夥人共同創立了樹莓派基金會。「樹莓」（Raspberry）這個詞是為了向其他以水果命名的公司致敬，例如 Apple、Acorn 等，而 Pi 象徵的是程式語言 Python。

此一研發中的設備在 2011 年 5 月已初步成形。一位 BBC 記者 Rory Cellan-Jones 將樹莓派的影片放到他高人氣的部落格上，僅僅兩天就獲得了 60 萬次的點閱率。因緣巧合之下，厄普頓對這 60 萬人許下承諾打造一個 25 美元的低價電腦。6 位合夥人投資了 25 萬美元（相當於製作一萬台機器的資金），本來計畫用幾個月的時間把產品（樹莓派）賣出去，沒料到在 2012 年初發表第一天就賣了 10 萬台樹莓派，到了 2014 年 10 月，更已賣出大約 380 萬台。2015 年 2 月，樹莓派基金會發行了第二代產品—樹莓派 2，售價 35 美元。樹莓派 2 採用 4 核 Broadcom 晶片、雙核 GPU 和 1GB 記憶體。

樹莓派本身即是一部小電腦，而與個人電腦最大的不同在於它有 GPIO 介面（General purpose input/output）可以連接各式各樣的感測器以偵測外部世界的狀況，並可透過 Python 程式設計將運算結果輸出，因此可支援機器人程式設計。此外，樹莓派搭載 Linux 可讓學生學習如何調校作業系統，而且它支援網路運作，例如學生可在實作網路專題時，透過外部感測器擷取網路數據，從中理解網路運作原理。學生亦可透過樹莓派專用的攝影機記錄外部影像，再經由 Python 與 OpenCV 函式庫處理，進行影像辨識與物體追蹤。這些實驗都有助於學生理解電腦科學內涵。

3. 結語

國小學生認知資源有限，所以我們要在有限的資源中給學生最精華的知識，這樣才能達到「Less is More」的效果。筆者所編的 Scratch 12 堂課正符合這個精神。如果我們只能給小學生少量知識，而這些知識又不夠精華的話，那就會變成「Less is Less」的狀況，學生只能學會很少的東西。因此，筆者建議在國小階段使用 Scratch，以案例式教學引導學生創作電腦遊戲，並應將程式設計概念與解題能力培養融入遊戲創作過程中，而不只讓學生按圖施工。

到了國中階段，應將程式設計納入 STEM 整合教育之中，結合 Scratch 與 Arduino 進行機器人程式設計。此外，筆者也建議讓學生在國中階段能開始接觸文字式程式語言，Greenfoot 是一個很好的選擇。在高中階段，筆者建議使用低價的硬體 Raspberry Pi 與自由軟體 Python，除了讓學生進一步增進文字式程式設計能力，也經由程式設計學習作業系統、網路等電腦科學概念。

如果我們希望高中生能夠達到「More is More」的境界，必須靠小時候有完整的知識骨架做支撐，否則表面上我們教給學生很多，但實際上他們卻學到很少，這就是「貪多嚼不爛(More is Less)」的現象。在中小學程式設計課程的規畫中，國小要做啟蒙教育，國中要做適性教育，高中要做普及教育，也要做菁英教育。這些都是筆者所關心的點。

參考文獻

- 石明豐(2015)。VPhysics。2015 年 12 月 13 日，取自：<http://vphysics.ntu.edu.tw>
- 高慧君(2013)。程式設計邏輯訓練—使用 Scratch。台北：松崗。
- 高慧君(2015)。Scratch2.X 用積木玩程式設計。台北：松崗。
- 廖祐梓(2012)。遊戲導向教材對高中生程式設計學習之影響。國立臺灣師範大學資訊教育研究所碩士論文。
- 賴鴻州(2014)。智高 Scratch(S4A) 互動智能積木：動手玩創意 20 堂課。台北：台科大。
- EET(2014)。樹莓派創辦人深入台灣「自造世代」。2015 年 12 月 13 日，取自：
http://m.eettaiwan.com/ART_8800703257_480102_NT_22fee30e.HTM#.Vm0IINJ97b0
- Congress(2015). H.R.1020 - STEM Education Act of 2015. December 13, 2015, from the World Wide Web:<https://www.congress.gov/bill/114th-congress/house-bill/1020>
- Kolling M.(2010). The Greenfoot programming environment. ACM transactions on Computing Education, 10(4). Article 14, 1-21.
- Kolling M.(2010). Introduction to Programming with Greenfoot. Upper Saddle River, New Jersey: Pearson.