

Scratch 教學程式模組設計初探

周文忠

國立屏東教育大學資訊科學系

chou@mail.npue.edu.tw

李俊霖

國立屏東教育大學資訊科學系

swps9804@yahoo.com.tw

摘要

本研究的主要目的是探討如何利用 Scratch 軟體之角色物件匯出及匯入的特性，依照教學需求，設計成不同功能屬性且獨立的程式模組，藉此提供 Scratch 程式設計的範例，方便應用於各項程式專案中，並將程式模組作為輔助國小學童學習 scratch 程式設計的鷹架，以降低學童對程式設計的恐懼及挫折感，增進其學習動機，促進自主學習，減少教師教學指導的負擔。

關鍵字： 程式模組、Scratch、程式設計

壹、緒論

隨著資訊教育向下扎根推展，教育部（2008）在「2008-2011 中小學資訊教育白皮書」中提倡多元學習、資源共享、創意提升的理念，並強調中小學教育是培養中小學學生資訊科技應用能力、態度與行為的教育。學習程式設計可以讓學生發揮創意巧思，建構自己的知識，建立解決問題的能力，並培養其自主學習、獨立探索及合作學習的態度，增強其資訊應用能力。

du Boulay(1989)指出初學者所使用的程式語言應該具有「簡單化」、「視覺化」兩大特性(引自 陳文鴻，2010)，並應具有物件介面導向與事件驅動機制，來降低學習程式設計之難度(賴健二，2004)。使用一般的專業程式語言進行程式設計時，初學者常會遇到缺乏親和力的程式介面、需記憶繁瑣複雜的英文指令語法、程式除錯判讀不易以及缺少開放程式原始碼可供參考的問題，這些過高的學習門檻，都容易影響學習者的學習意願，降低學習動機，甚至放棄學習(何昱穎、張智凱、劉寶鈞，2010)。

研究者擔任國小資訊課程教師，2007 年 Scratch 軟體推出後，開始以 Scratch 軟體教導學童設計程式，學生反應相當良好。Scratch 是針對八歲以上的兒童所開發的程式語言，利用視覺化的設計介面，將繁瑣複雜的程式語言指令轉換成個別的積木，使用者只需要透過拖曳、組合各類型積木，依角色需求設定不同的參數，就可以完成各式各樣的多媒體故事、動畫、音樂創作及遊戲。目前

的版本軟體介面也完全支援繁體中文，對於國小學童而言，更沒有語言的隔閡；Scratch 軟體可免費下載，取得方便，不用擔心付費及使用權的問題；更值得一提的是開發小組設立的官方網站，支援多國語系，提供世界各地的 Scratch 程式設計者申請註冊，註冊後即可公開自己的 Scratch 專案作品上傳至網站供瀏覽者欣賞，也可以下載別人的專案程式進行交流研究，所有的專案都是開放程式設計原始碼，解決了缺少程式設計原始碼可供參考學習的缺點。

Scratch 是秉持建造主義的理念設計而成，經由做中學的程式設計過程，將抽象的經驗轉換為具體的程式來呈現。Scratch 雖然已改善了前述的學習程式設計的門檻，但是研究者在實際教學的過程中，觀察發現多數初學 Scratch 程式語言設計的國小學童會有不知如何運用正確指令積木及參數設定等邏輯思考方面的問題，除了必須花費很多時間指導外，也增加了學童對程式設計的恐懼及挫折感。

經分析高雄市 Scratch 程式競賽的國小學童參賽作品內容，發現多數小朋友喜歡設計電流急急棒和射擊類的遊戲程式。本研究的目的即針對這兩種類型遊戲經常使用的指令積木，藉由 Scratch 軟體裡角色物件匯出及匯入的功能，著手設計一系列可應用於這兩類遊戲程式設計的程式模組。將程式模組化後具有標準化和分割性的特點，物件之間可以彼此相容，並隨任務需要而進行組合、拆解，可以重複使用 (Ed Peschko et al., 2001)。D. L. Parnas (1972) 亦認為程式模組化，具有縮短管理發展的時間，使程式更具有彈性，容易被瞭解的優點 (引自 張宇華，2003)。

因此，本研究以 Scratch 程式模組當作學生學習 Scratch 程式設計時的學習輔助鷹架，藉由程式模組提供邏輯思考的範例，期能改善上述初學者常遭遇的問題，並實際運用於程式設計的專案中，以達到輔助學習的成效。

貳、文獻探討

一、Scratch 程式語言

Scratch (<http://scratch.mit.edu/>) 是由美國麻省理工學院媒體實驗室的米契爾·雷思尼克 (Mitchel Resnick) 教授所領導的終生幼稚園 (Lifelong Kindergarten) 團隊開發完成，於 2007 年首度公開，目前的版本為 1.4 版，具有繁體中文程式介面。現行版本的 Scratch 也支援 Arduino、PicoBoard 與樂高 (LEGO) WeDo 等外接式模組，也可以透過 Kinect 與 Scratch 的中介程式，與微軟 Xbox 360 之體感控制器 Kinect 做結合，增加了 Scratch 在教學上的互動性 (許惠美，2011)。Scratch 讓數位表達更加大眾化，透過 Scratch 的軟體，小朋友們可以創造他們自己互動的故事、遊戲以及動畫，並且可以在線上分享他們的創作。當小朋友在學習重要的電腦程式運算，同時他們亦學會創意性思考、有系統地推理以及協同合作，Scratch 已經被稱之為互動媒體的 Youtube (Resnick, 2011)。

Scratch 的程式設計專案，包含了媒體元件(media)和程式腳本(script) 兩大部分。Scratch 可以從外部匯入圖片檔及聲音檔加以運用，也可以利用內建的繪圖工具及錄音功能來製作所需的媒體元件，並透過彩色的指令積木，針對舞台(stage)和所選定的 2-D 角色(sprite)進行程式腳本的設計(Resnick et al.,2010)。整個 Scratch 是以降低新手門檻、增加專案的種類以及支持不同的專案複雜度為目標，來進行規劃環境介面(Resnick et al.,2009)。因此撰寫程式腳本就是以指令積木為物件基礎，透過滑鼠，將需要的指令積木從拖曳至程式腳本區堆疊拼貼，整個 Scratch 程式碼就是由指令積木加上參數設定所組合而成，如圖 1。

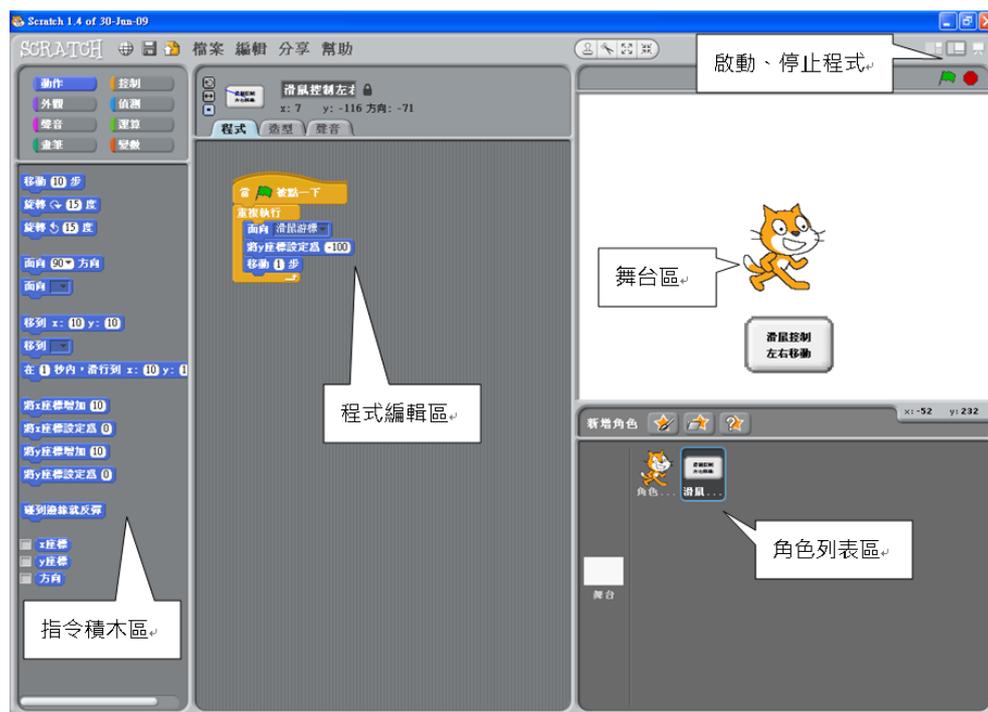


圖 1 Scratch 環境介面

Scratch 在開發之初，即是以兒童及青少年為主要的使用對象，而他們大都是程式設計的初學者，在考量其先備知識不足與能力有限的情況下，因此 Scratch 有許多與一般程式語言迥然不同的創新設計，Scratch 具有以下的特色：

- (一)屬於免費軟體，可擴充硬體支援。
- (二)具有視覺化的單一視窗及多國語言介面。
- (三)積木組合式程式語言，容易學習。
- (四)內建豐富的圖片及音效素材，亦可自行製作素材並支援素材匯入的功能。
- (五)直觀式編譯，執行結果立即可見，除錯容易。
- (六)公開專案程式碼，有助於觀摩學習，並可上傳作品至官方網站與全世界同好分享。

而 Scratch 程式語言具有直覺式的中文介面設計，互動性高，並用拖曳指令積木取代了傳統編寫語法指令，並將程式碼包含在角色物件中，使得在程式設計過程中，可以重複使用這些物件，執行時所見即所得，這些直觀視覺化的設計，大幅降低學習的困難度，也提高了學生的學習興致。學習過 Scratch 程式語言的學生，大都持正向的學習態度，表示喜愛 Scratch 程式設計課程並認為是一種有

趣的活動，不會感到焦慮或排斥(楊書銘，2008；王麒富，2009；蔡孟憲，2010；張素芬，2010；蕭信輝，2010)。研究結果亦顯示實施 Scratch 程式設計課程，對於國小學生的問題解決能力有顯著的提升(王麒富，2009；蕭信輝，2010；賴明宏，2010)，對幾何概念的學習亦有顯著的影響(蔡孟憲，2010)。

二、鷹架理論

蘇聯心理學家維高斯基(Vygotsky,1978)的認知發展論中，提到介於兒童自己能力所能達到的水平，與經別人給予協助後所可能達到的水平，這之間的成長空間，就是所謂的可能發展區(zone of proximal development)，經由外在的因素來促進及協助兒童的發展成長，即稱為鷹架 (scaffolding)(張春興, 1994)。

在鷹架理論中，學習鷹架是一種輔助兒童學習的概念，以及完成學習過程的適當程序。適當的鷹架行為能夠輔助兒童更清楚地了解學習內容，並得以順利達成學習目標，也是教師進行教學時的重要原則。也就是說，透過鷹架行為逐漸改變兒童對學習內容的了解程度，因此，隨著兒童能力的成長發展，對於引導兒童完成相同學習內容所需要的鷹架只會愈來愈少。張苑珍(1997)認為鷹架行為重視學生主動、積極的學習，屬於一種暫時性的支持，必須視學習情況而改變上架或撤架。

三、程式設計相關研究

剛開始接觸程式設計的學習者，最常遭遇的困難主要是程式設計的概念抽象，學習程式主要有語法指令和演算思考兩個階段(豐佳燕、陳明溥，2008)。而進行演算思考，需將問題描述清楚，細分成多個較好處理的小問題，再逐步解決每一個小問題(Stephen Copper、Wanda Dann、Randy Pausch，2000)。因為這些邏輯思考程序本質不易理解，所以初學者使用的程式語言應該具有「簡單化」、「視覺化」兩大特性(du Boulay，1989 引自陳文鴻，2010)。若程式語言具有物件介面導向與事件驅動機制，更可降低學習程式設計之難度(賴健二，2004)。

張嘉宏(2007)的研究中指出學生學習程式語言後，卻仍然不具備程式設計的能力，主要的問題可歸類為以下三類：

(一)程式語言的複雜觀念

傳統的程式語言強調其語言設計的專業功能，不重視人機互動介面，初學者在學習程式語言時容易因語法過於複雜、觀念不清與錯誤信息難以理解等造成學習進度緩慢，甚至放棄學習的問題。對於程式語言的初學者而言，選擇一個互動程度高而且又能夠以簡單的語法，來訓練初學者邏輯思考及程式設計能力的程式語言，對提升程式語言的教學效益來說，便非常的重要。(董少桓、林宗德、沈維倫、錢傳明、沈勇嘉，2005)。

(二)老師需要付出很大的教學心力

傳統程式教學時，學生如有問題大都只能透過提問或自習的方式來解決學習上的問題，老師亦只能靠經驗判斷、個別指導或作業中來瞭解學生的學習情形。老師在準備教材、批改學生的程式或是要解決學生個別不同程度的問題，都

需要花費很多的時間與心力，也影響學生的學習效率。

(三)學生缺乏實際練習

一般程式設計的課程，教師若採用講述式教學，學生不易在短時間內理解學習，加上學生對於老師指派的程式設計作業練習不確實，都會影響學習成效。如果學生能夠在學習的過程中，透過一個簡易好用的介面來練習程式，透過實作來學習知識(learning by doing)，利用漸進的方式來引導學習，可以幫助學生迅速的學習程式技巧、解決問題與邏輯思考的能力(錢傳明，2008)。

黃嘉文(2010)的研究指出，建造式語言系統，雖然使用圖形化或實體物件呈現程式的架構，讓學習者在系統中建造出程式片段過程中，明白不同物件搭配所產生的結果，進而從中瞭解程式語法和命令的意義。但卻忽略了程式設計，事實上是一種抽象化且難以理解的問題，若在設計主題採用了與學習者本身未擁有的經驗與知識，學習者將會缺乏從經驗中將新舊知識融合的過程，容易因失敗經驗而產生挫折感，因而降低學習興致。

從研究文獻中，發現初學者在學習程式語言的過程中，會參考別人的範例程式，從中參考模仿學習；如遇到新的問題時，則會參考功能類似的程式碼，設法修改調整功能來解決問題。在程式設計問題解決過程中，如果能提供適當的程式設計工具，幫助學習者將問題視覺化或物件化，將能促進學習者的思考與解題(Funkhouser, 1993)。

完整的程式可說是一群模組的集合，程式模組是大程式中邏輯獨立和離散的單位(Anthony Ralston et al., 2000)。模組化(Module)程式設計的觀念，乃是把程式設計的問題由大至小，從上而下，進行分析後分成不同層次的片段程式，形成各自獨立的程式模組物件。模組化具有標準化和分割性兩個特點，物件之間可以彼此相容，可以隨任務需要而進行組合、拆解，可說是物件導向(Object-Orientation)技術的前身，具備物件導向 Has-A(Reuse-重複使用)的精神，但不具備 Is-A(Inheritance-繼承)的概念(Ed Peschko et al., 2001)。D. L. Parnas(1972)認為程式模組化，包含了可以縮短管理發展的時間，使程式更具有彈性，容易被瞭解的優點(引自 張宇華，2003)。

參、Scratch 教學程式模組設計

一、Scratch 程式模組架構

本研究所設計的程式模組，是依不同功能需求將 Scratch 指令積木群組合成一段可單獨執行的程式碼依附在角色裡，採用物件方式，利用 Scratch 程式語言的角色匯出及匯入的功能，將程式碼運用至不同的專案上，程式設計者只要程式碼複製至新的角色或是直接變更角色造型，再適當調整參數，即可在新的專案上使用該程式模組，Scratch 程式模組架構，如圖 2。

設計 Scratch 教學程式模組，其程式架構應包含以下三個要素：

(一)指令積木：依功能需求，將各類別的指令積木組合成符合程式邏輯的程式碼片段。

- (二)變數及參數：變數的用途在於儲存資訊，參數的用途則是調整及控制程式的進行，使其符合需求。
- (三)註解：將指令積木、變數及參數等程式碼的內容加以解釋說明，作為學習者程式設計及應用時的參考。

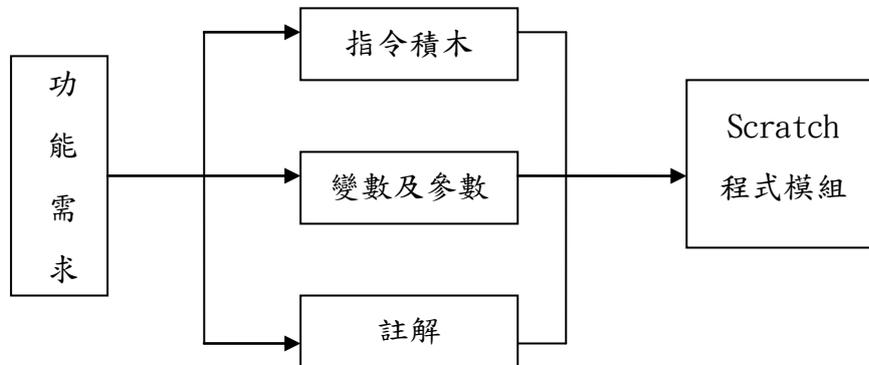


圖 2 Scratch 程式模組架構圖

二、Scratch 教學程式模組設計範例說明

程式模組設計時應掌握問題由大至小，從上而下的原則，使其具有標準化和分割性的特性。

以「向前計時」的 Scratch 程式模組為例，如圖 3。以下將整個模組分為指令積木、變數及參數、註解三個部分，分別就其設計重點加以說明。



圖 3 Scratch 程式模組範例-E02 向前計時

(一)指令積木：

1. 瞭解功能需求後，依問題分析指令積木的種類，本範例需使用控制類和變數類的指令積木。
2. 每段程式皆以波浪型積木作為開始，以平底型積木作為結束。
3. 指令積木的組合需符合程式運作的邏輯，依序堆疊嵌合成有意義的程式碼。

(二)變數及參數：

1. 分析使用的變數項目，並設定變數名稱，Scratch 允許使用中文變數名稱，學習者可以更容易瞭解，本範例需設定「時間」變數。
2. 設定變數的初始值，如將時間變數初始值設為 0。
3. 設定各項參數值，並加以調整以符合功能需求。

(三)註解：

1. 教學程式模組強調輔助學習鷹架的作用，所以註解應標明模組用途、參數、變數名稱以及指令積木的功能，讓學習者容易判讀。
2. 依照程式運作邏輯加以說明，使學習者瞭解程式的運作步驟及原理。

三、Scratch 教學程式模組種類

本研究依照需求功能不同，將 Scratch 教學程式模組分為五大類，共計 24 種程式模組。

(一)移動控制類

程式模組編號 A01~A07，其功能包含：滑鼠控制角色移動、滑鼠控制角色上下移動、滑鼠控制角色左右移動、角色緊靠滑鼠游標移動、鍵盤控制角色左右移動、鍵盤控制角色上下移動、鍵盤控制角色四方移動，共 7 種程式模組。

主要功能是利用滑鼠或鍵盤的按鍵來控制角色移動，適合應用在玩家操控的角色使用，可修改移動距離參數決定角色移動的速度。

(二)自動移動類

程式模組編號 B01~B03，包含：角色自動移動如碰到邊緣就反彈、角色依方向自動移動如碰到邊緣就反彈、角色自動移動如碰到邊緣就反彈並旋轉，共 3 種程式模組。

主要功能為讓角色會自行移動，碰到邊緣後會自動反彈或旋轉反彈往回走，適合應用在電腦程式控制的角色使用，可修改參數決定角色移動的速度及旋轉的角度。

(三)射擊類

程式模組編號 C01~C03，包含：角色(子彈)水平方向發射、角色(子彈)垂直方向發射、角色(子彈)發射至滑鼠游標位置，共 3 種程式模組。

主要功能為按下發射鈕後角色會自動依照滑鼠的方向或是程式所設定的方向、速度前進，直到碰到設定的條件或是邊緣才會消失停止，適合應用在射擊類遊戲的物體發射攻擊使用，可修改參數決定角色移動的方向、速度。

(四)特效類

程式模組編號 D01~D08，包含：亂數隨機決定出現座標、角色自動變色、讓角色淡入逐漸出現、讓角色淡出逐漸消失、讓角色的外型逐漸變大、讓角色的外型逐漸變小、切換角色外型產生動畫效果、碰到設定之條件即消失隱藏，共 8 種程式模組。

主要功能包含角色的出現由亂數隨機決定位置、改變外型顏色、出現及消失的方式、外型的大小變化、外型切換以及碰觸條件後消失等相關程式模組，適

合應用在角色外型、碰觸條件設定及出現位置變化的使用，可修改參數決定角色變化的狀況。

(五)計時類

程式模組編號 E01~E03，包含：向前計時直到條件滿足停止、持續向前計時、倒數計時直到條件滿足停止，共 3 種程式模組。

主要功能包含向前計時和倒數計時，可修改參數決定計時開始及結束的條件，適合應用於遊戲的時間條件設定使用。

肆、結論及建議

綜觀以上的論點，以 Scratch 程式模組進行教學，預期可以達到以下的功能：

(一)提高程式設計的效率：

將程式模組化是把一個大程式，分成若干小程式，相同功能的程式片段可以在不同的程式設計專案重複使用，應用上具有彈性，不僅節省記憶體空間，並可以減少程式開發的時間。

(二)提高程式的可讀性：

程式模組化可以縮短程式的長度，簡化程式的邏輯。且每一個模組皆加上中文註解，讓學習者更容易瞭解程式的寫法及運作。

(三)除錯及維護容易：

採用模組化的設計，不同功能的程式模組單元分割且獨立，可有效減少除錯及維護的時間及成本，使得程式模組應用在不同的專案時，可以依需求修改或調整後立即使用。

(四)成為程式設計的鷹架：

Scratch 教學程式模組是依照不同功能設計而成，學生可以程式模組作為學習的鷹架，來協助進行程式設計。隨時可依個人的學習狀況及設計需求而調整使用，或是以原有模組為基礎，發展出新功能的模組，擴大其應用範圍。

(五)減輕教師教學負擔：

教學程式模組除了可以直接應用到程式專案中，並可當作程式設計教學的範例，有助於學生於自主學習。

(六)增強學習動機：

運用程式模組進行 Scratch 程式設計，可以加快程式完成，降低程式邏輯的錯誤，避免因失敗經驗對程式設計產生恐懼，促使學習者從過程中獲得成就及滿足感，增強其學習動機。

本研究目前只針對電流急急棒及射擊遊戲類遊戲需求開發設計程式模組，未來可依不同的教學內容發展其他的 Scratch 程式模組，擴展其應用範圍。

伍、參考文獻

- [1]王國川(2008)。國小中年級學生以 Scratch 學習程式語言設計之研究。佛光大學學習與數位科技學系碩士論文。
- [2]王曉璿、王麒富、林建伸(2009)。應用直觀式 Scratch 軟體輔助國小學童問題解決合作學習教學設計初探。GCCCE2009 第十三屆全球華人計算機教育應用大會。973-976
- [3]王麒富(2009)。應用直觀式 Scratch 軟體提升國小學童問題解決能力效益之研究。國立台中教育大學數位內容科技學系碩士論文。
- [4]何昱穎、張智凱、劉寶鈞(2010)。程式設計課程之學習焦慮降低與學習動機維持 - 以 Scratch 為補救教學工具。數位學習科技期刊, 2, 1, 11-32。
- [5]吳聲毅(2009)。數位學習觀念與實作第二版。台北:學貫行銷。
- [6]李長駿(2009)。一套註記工具支援教學策略來改善學童學習程式語言之學習成效。國立中央大學資訊工程研究所碩士論文。
- [7]張宇華(2003)。模組化知識管理系統之建構。樹德科技大學資訊管理研究所碩士學位論文。
- [8]張春興(1994)。教育心理學-三化取向的理論與實踐。台北:東華書局。
- [9]張素芬(2010)。國小資訊教育實施 Scratch 軟體教學之研究。高雄師範大學工業科技教育學系研究所碩士論文。
- [10]張苑珍(1998)。鷹架理論在成人教學實務之應用。成人教育, 第 40 期, 43-52。中華民國成人教育學會。
- [11]張嘉宏(2007)。JSP 教學系統設計與評估。國立雲林科技大學資訊管理系碩士論文。
- [12]教育部(2008)。教育部中小學資訊教育白皮書 2008-2011。教育部。
- [13]許惠美(2011)。美國初等教育中建造主義實踐之初探。網路社會學通訊第九十七期。南華大學社會學研究所。
- [14]許惠美(2011)。淺談 Scratch 與 Kinect 結合與實作方法—讓你的 Scratch 更具互動性。宜蘭縣教育資訊電子週報第 34 期。
- [15]陳文鴻(2010)。運用概念圖輔助國小學童主題式程式設計課程效益探究。第六屆科技教育課程改革與發展學術研討會 (ICCITE 2010)國內研討會論文。
- [16]黃嘉文(2010)。以 Cyber-Physical 環境支援程式設計學習之探究。國立中央大學網路學習科技研究所碩士論文。
- [17]楊書銘(2008)。Scratch 程式設計對六年級學童邏輯推理能力、問題解決能力及創造力的影響。臺北市立教育大學數學資訊教育教學碩士論文。
- [18]董少桓、林宗德、沈維倫、錢傳明、沈勇嘉(2005)。支援教師分身的智慧型程式言教學平台。雲林科技大學資訊管理系。
- [19]蔡孟憲(2010)。Scratch 程式設計對國小五年級學生幾何概念及邏輯推理能力的影響。臺北市立教育大學數學教育資訊學系教學碩士班碩士論文。

- [20]蕭信輝(2010)。Scratch 程式設計對國小五年級學童科學過程技能、問題解決能力及後設認知之影響。台北市立教育大學數學教育資訊學系教學碩士班碩士論文。
- [21]賴明宏(2010)。Scratch 程式對國小五年級學童邏輯推理能力與科學問題解決能力影響之研究。國立臺北教育大學自然科學教育學系碩士班碩士論文。
- [22]賴健二(2004)。兒童視覺化中文程式語言之開發與研究。國立台北師範學院教育傳播與科技研究所碩士論文。
- [23]錢傳明(2008)。支援同儕學習的程式語言教學平台。雲林科技大學資訊管理系碩士論文。
- [24]豐佳燕、陳明溥(2008)。國小學生學習電腦程式之研究- 以 Stagecast Creator 創作遊戲為例。GCCCE2008 第十二屆全球華人計算機教育應用大會。
- [25]About Scratch(2011). http://info.scratch.mit.edu/About_Scratch
- [26]Anthony Ralston, Edwin D.Reilly, David Hemmendinger, 2000, Encyclopedia of Computer Science, 4th ed., Nature Publishing Group.
- [27]D. L. Parnas(1972). “On the Criteria To Be Used in Decomposing Systems into Modules” , Communications of the ACM, Vol. 15, No. 12, pp. 1053 –1058, December.
- [28]Design Envy (2011), Scratch: MIT Media Lab, <http://designenvy.aiga.org/scratch-mit-media-lab/>
- [29]du Boulay, B., O'Shea, T. & Monk, J. (1989). The Black Box Inside the Glass Box: Presenting Computing Concepts to Novices.Studying the Novice Programmer. E. Soloway and J. C. Spohrer.Hillsdale, NJ, Lawrence Erlbaum Associates: 431- 446.
- [30]Ed Peschko & Michelle Dewolfe & Edward S. Peschko,2001,Perl 5.6 Complete,McGraw-Hill Education.
- [31]Funkhouser, C. P. (1993). The influence of problem solving software on student attitudes about Mathematics. Journal of Research on Computing in Education, 25(3), 339-346.
- [32]Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, 52(11), 60-67.
- [33]Resnick,M.(2011).<http://web.media.mit.edu/~mres/>
- [34]Resnick,M., Maloney, J., Rusk, N., Silverman, B., & Eastmond, E.(2010). The Scratch Programming Language and Environment. ACM Trans. Comput. Educ. 10, 4, Article 16.
- [35]Stephen Copper, Wanda Dann, Randy Pausch (2000). Alice: A 3-D Tool for Introductory Programming Concepts, JCSC, the Consortium for Computing in Small Colleges, 108-117.