

# 運用視覺化程式設計工具 降低學習程式設計之焦慮與提高學習動機

何昱穎

國立臺南大學數位學習科技學系

Email: yuying@mail.nutn.edu.tw

張智凱

國立臺南大學數位學習科技學系

Email: chihkai@mail.nutn.edu.tw

**摘要**—本研究以修習 C++ 程式設計的大一學生 45 位為研究對象，探討學習成就、學習焦慮與程式語言之間的關係。首先在期中考後使用學習焦慮量表施測，結果顯示學習成就低的學生學習焦慮較高，尤其以學習成就在全班四分之一以下的 12 位學生最為明顯；其次針對這 12 位學生，以 Scratch(一種圖形化程式語言)進行補救教學；補救教學後收集學生對 C++ 與 Scratch 的有趣性(Playfulness) 與愉悅性(Enjoyment) 的問卷反應，結果顯示 Scratch 顯著較高，補救教學後這 12 位學生的學習焦慮明顯降低；分析學生補救教學前後之成績，也以這 12 位學生的成績進步最為明顯；最後對這 12 位學生進行非正式訪談並分析補救教學中的 Scratch 作品。整體而言，研究發現使用 Scratch 能在悅趣的氣氛下進行有效的補救教學，並提高學生成績，值得程式設計教師參考採用。

**關鍵詞**—程式設計、學習成就、學習焦慮、補救教學、Scratch

## Using visual programming language to reduce learning anxiety and keep learning motivation of programming course

*Abstract* — This study explores the relationships among learning achievement, learning anxiety, and computer programming language for novices by using visual programming language, such as Scratch, for remedial instructions. The subjects (n=45) were freshman who took the C++ programming design course. “Anxiety of Learning Computer Programming” and “Playfulness and Enjoyment of Learning Computer Programming” questionnaires were used as the survey tools. The collected data was analyzed by descriptive statistics and the semi-structured interview. The experimental results show that the remedial instruction of using Scratch can reduce student's computer programming anxiety, and keep student's learning motivation of computer programming. The remedial instructions can improve learning achievement, especially for students with lower performance (n=12). Consequently, using Scratch for remedial instruction is recommended for instructors of computer programming.

*Keywords:* computer programming, learning achievement, learning anxiety, remedial instruction, Scratch.

## 一、簡介

大部分初學者對程式語言的第一印象是由英文字母和一堆陌生的符號所組成，學習者開始撰寫程式解決問題時，常常不知從何下手，即使做過許多練習，對於程式設計的工作還是感到困難，想成為專業的程式設計師，通常需要經過多年的訓練。程式設計在電腦科學的專業領域中是一項重要的基礎能力，但台灣目前國小到高中的電腦科學課程大多以應用軟體及網際網路教學為主，主要強調應用面向，三個階段的授課內容重複性高，學生未能普遍學習有助於培養邏輯思考與創新思維的課程內容；而大學則是資訊相關科系才藉由學習程式設計來了解電腦科學的運作理論，通常這類課程會選擇一種程式語言，如 C++ 或 Java，作為入門的程式語言，對大一的程式初學者而言，不一定能立刻適應課程所教授的程式語言。

以 C++ 或 Java 而言，都是設計來支援大型軟體計畫的開發，解決產業界實際的複雜問題，需要編譯後才能執行，IDE 包含許多初學者不會使用的功能。學習這類程式語言除了需要相當的邏輯思考、數學演算及抽象思維能力，所需的認知負荷，如關鍵詞、保留字及區塊符號等也很高，這往往導致許多學習障礙。而使用 C++ 或 Java 這類專業、互動程度低的程式語言來教學，初學者會產生如語法複雜、錯誤訊息難以理解、學習進度緩慢等許多妨礙學習的問題 (Felleisen, Flinger, Flatt, & Krishnamurthi, 2004)，影響學習成就，這可能減低學習有趣性與愉悅性，造成學生學習焦慮。

因此先選擇一個互動程度高且語法簡單的程式語言來訓練初學者的邏輯思考及程式設計能力，可提升程式語言教學效益。對初學者而言，先透過高度互動，迅速的學習邏輯思考能力並解決問題，比學習複雜的語法與開發工具要來的重要 (董少桓、林宗德、沈維倫、錢傳明和沈

勇嘉, 2005)。由 MIT (麻省理工學院) 所發展的一種圖形化程式語言 Scratch，以類似積木的拼裝方式，把各類物件組合在一起完成程式設計，非常容易上手，非常適合訓練初學者。

本研究以修習 C++ 程式設計的大一學生為研究對象，探討學習成就、學習焦慮與程式語言之間的關係，在期中考及補救教學後皆使用學習焦慮量表施測，並分別收集學生對 C++ 的有趣性 (Playfulness) 與愉悅性 (Enjoyment) 的問卷反應；研究選擇以 Scratch 對全班進行補救教學，分析學生期中考 (補救教學前) 及期末考 (補救教學後) 之成績以探討對學習成就的影響；並針對低成就組 (期中考成績在全班四分之一以下) 的 12 位學生進行非正式訪談與分析補救教學中的 Scratch 作品，以了解影響學習成就的要素。

## 二、文獻探討

補救教學是一種診療式教學，是一種「評量—教學—再評量」的循環歷程 (杜正治, 1993)，可幫助學生學習過去認為困難的課程內容，幫助學生發展並整合學習策略 (Julie & Ruby, 2006)。補救教學的對象是學習低成就的學生，是為了診斷低成就學生學習困難的原因所採取補救的學習措施 (高廣孚, 1988)，低成就學生的定義為智力正常，但實際學業表現明顯低於應有的能力水準，可細分為三類：一是學生的實際學業表現明顯低於應有的能力水準；另一是學生的實際學業表現明顯低於班級平均水準；最後一類是學生學科成就不及格，且學業成就明顯低於其他學生許多，稱為成績低落 (張新仁, 2001)。

補救教學透過診斷、補救與評估程序，從學生的學習困難處著手，提供學生適當的指引，使學生有機會克服困難達到學習目標，期望補救教學實施一段時期後，低成就學生能夠跟上全班的進度 (黃淑苓, 1999)。因此補救教學應選擇好接受補救教學的對象後，再進行教學，在了解學生的學習困難後，應精心設計課程內容與慎選教學

型態與策略，才能確實符合學生的需求(邱上真，1992)。補救教學的課程內容設計首先要考慮由易至難、由簡而繁、從已學到未學等學習原則，才能建立學生的自信心並提高學習動機；需要訂定具體明確的單元學習目標，學生才能掌握學習重心；學習活動的設計要考慮學生的能力、接受程度及注意廣度。對低成就學生而言，教材應簡化，學習活動應更富有變化，更具趣味性(張新仁，2000)，常用的補救教學課程內容設計有補償性、導生式、適性、補充式、加強基礎性及學習策略等類型(杜正治，1993)。

國內外常用的補救教學型態包括資源教室、學習站、學習實驗室、套裝學習材料及電腦輔助教學。電腦輔助教學是一種利用電腦呈現教材、控制教學進度的教學模式，不論學生的程度、能力、動機或態度，只要投入學習，電腦會提供立即的回饋，若學生做出正確的反應，電腦會立即提供獎勵，提高信心，電腦輔助教學的操作方式簡便，易記易學，電腦軟體可針對學生個別需求設計，做大量練習，用途廣泛，學生可自訂進度，依照學生能力與程度，循序呈現新的教材(張新仁、邱上真、李素慧，1999)。

國內外常用的補救教學策略包括直接教學法、合作式學習、精熟教學及個別化教學。精熟教學的基本理念是每個人的學習速度快慢不同，強調教師必須擬定特別的教學目標，讓學生了解整個教學的程序，使學生循序漸進練習，教師列出要求學生精熟的標準，只要給予學生足夠的學習時間，幾乎所有智力正常的學生，都能精熟大部份學習內容，教師必須於每個單元結束後舉行診斷測驗，對學生的錯誤及學習困難給予立即的回饋，精熟教學法可以使遲緩及缺乏動機的學生變的有效率，充滿學習自信。學生因而改變學習情緒，獲得基本技能，有助於學習(張新仁，1995)。

焦慮是一種對情境感到緊張或憂心的感覺，根據學習理論的說法，焦慮是一種心理壓力所引

發的結果，通常是因缺乏明確目標而產生不安焦躁的情緒。焦慮可分為個人潛在因素、個人態度、老師態度、與老師互動關係、班級氣氛及考試焦慮六類，其中焦慮與考試成績成負相關(Young, 1991)，適當焦慮可幫助學生提高學習效果，使成績進步，但高度焦慮就可能產生學習上的抑制，降低學習成就(張春興，1999)。學習態度、學習動機與學習焦慮具有強烈的關聯性，是影響學習成就的重要因素，愈正向的學習態度會有愈好的學習成就，也就是說，愈高的學習焦慮會導致愈負向的學習態度，若降低學習焦慮、提高學習動機將有助於提高學習成就(林震城，1997)。對程式學習者而言，若無法理解程式或寫出程式解決問題，造成學習成就低，則容易產生高度學習焦慮，當學習者對程式設計存在學習焦慮時，在學習程式設計或預期要接觸程式設計時，就會感到壓力，形成不安、緊張、恐懼或厭惡等情緒，這些情緒會影響學習者對程式設計的學習動機，學習者會出現避免和程式設計接觸、減少學習程式設計時間、進程式設計時極端小心或對程式設計存有負面想法等學習態度(Maurer & Simonson, 1984)，如此一來將形成負面循環。

初學者習慣使用高等、複雜的人類語言，導致無法運用指令少、語法固定的程式語言解決問題(林振輝，2003)，而程式設計包含許多解題的內隱歷程，這些歷程是解決問題的重要關鍵，初學者往往不易察覺(Kirsner, 1998)。先思考解決問題的流程，畫出流程圖後再轉換成程式語言或先學習程式語言的指令、語法，接著思考如何明確描述執行或中止程序行為的演算法，這兩種學習程式設計的主要方式對初學者而言都很困難。初學者遇到不知從何下手的程式時，可能根本不知如何畫出流程圖，利用流程圖雖可了解程式的運作邏輯，卻無法有效幫助初學者寫出程式(Winslow, 1996)，為幫助初學者更有效發展程式設計能力，在程式語言教學上，除了解程式語言

的基本結構，更須能有效運用程式語言解決問題 (Govender & Grayson, 2006)。程式語言的教學方式將影響問題解決能力的培養，程式設計工具的選擇，也會對初學者的程式設計學習產生影響，視覺化開發環境提供學習者更具象化的介面，可幫助初學者掌握程式設計的架構 (Brusilovsky & Spring, 2004; Shanmugasundaram, Juell, Groesbeck, & Makosky, 2006)。因此本研究將探討學習成就、學習焦慮與程式語言之間的關係。

### 三、研究方法與實驗設計

#### (一) 研究問題

綜觀有關補救教學的文獻可以發現，不管是何種課程內容設計、教學型態與策略，補救教學對低成就學生的學習效能都有正面的影響，學習有趣性對學習愉悅性有正面影響，學習有趣性與學習愉悅性對降低學習焦慮也有正面影響，整理研究模型如圖 1 所示 (Venkatesh, 2000; Yi & Hwang, 2003)。本研究僅探討其中四個研究假設，包括使用 Scratch 進行補救教學對學生學習焦慮的影響如何？對何種學生能最有效降低學習焦慮？使用 Scratch 進行補救教學對學習程式設計的有趣性與愉悅性的影響又如何？此外，利用學生作品分析探討使用 Scratch 進行補救教學時學生最常使用的程式設計概念以及是否與需補救的課程內容相符合？

- H1: 程式語言對降低學習焦慮有正面影響。
- H2: 程式語言對學習有趣性有正面影響。
- H3: 程式語言對學習愉悅性有正面影響。
- H7: 降低學習焦慮對學習成就有正面影響。

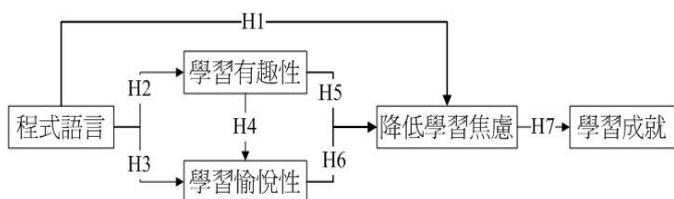


圖 1 研究模型

#### (二) 研究工具

##### 1. 學習焦慮量表、有趣性(Playfulness)與愉悅性(Enjoyment)問卷

為了探討學習成就、學習焦慮與程式語言之間的關係，研究者參考 Venkatesh 發表於 ISR 2000 的問卷設計一份學習焦慮量表以及有趣性與愉悅性問卷 (Venkatesh, 2000)，了解以 Scratch 進行補救教學後對學生 C++ 的學習焦慮、有趣性與愉悅性是否有影響。問卷共 19 題，分為學習焦慮、有趣性與愉悅性等三部份，學習焦慮量表問題有 9 題，其中 5 題為反向題，有趣性相關問題有 7 題，其中 4 題為反向題，愉悅性相關問題有 3 題，以 Likert 5 點量表問題來衡量，每題由非常同意到非常不同意共計 5 個選項，依序由 5 分到 1 分予以給分，反向題則反序給分。問卷在期中考後實施前測，以 Scratch 進行補救教學後實施後測。

##### 2. Scratch

Scratch 是由麻省理工學院媒體實驗室的 Lifelong Kindergarten 小組和加州大學洛杉磯分校的 Yasmin Kafai 小組合作發展的新一代程式語言，設計靈感來自於 Logo，並取代 Logo 的編碼程式。Logo 是 MIT 的 Seymour Papert 教授於 1960 年代所發展，主要設計理念在提供一個數學學習環境，讓學生在使用 Logo 程式設計過程中自由探索並學習數學概念，基本上可分為文字式語言 MSWLogo (Microsoft Windows Logo) 與圖像式語言 Drape (Drawing Programming Environment) 兩種版本，都很容易學習和掌握。Logo 語言不分大小寫，提供簡碼指令，兩個英文字母就可構成一個指令，使用 FD (Forward)、BK (Back)、LT (Left)、RT (Right)、PU (Pen Up)、PD (Pen Down) 及 RP (Repeat) 等指令，指揮一隻小烏龜在電腦螢幕上繪製許多複雜圖形。Logo 以程式設計者的觀點來解題，利用前進、後退、左轉、右轉等日常用語，十分容易把自己想像成是畫面中的小烏龜來完成工作，將學習與輸入指令所構成的學習

障礙降到最低，除了可繪畫圖案，還可計算和發出聲音，也提供重複結構、條件控制和副程式等較為進階的程式組成元素。

Scratch 則是一種視覺化程式語言，介面主要由舞台、角色列表、程式區、程式區塊等四個部份組成，如圖 2 所示，舞台為展示程式區成果的地方，舞台上按鈕可控制舞台模式，讓使用者展示已完成的作品；角色列表顯示舞台所有角色的縮圖，點選相對應的角色，選擇編輯標籤，可在程式區修改程式碼、造型及聲音，可以載入不同角色，也可以自行繪製角色的各種造型；將程式區塊中的指令拖曳至程式區進行組合來編輯程式碼，每個角色都有自己的程式區，可以有多段程式碼，Scratch 會自動分配時間來執行各段程式，如圖 3 所示，以類似積木的拼裝方式，把各類物件組合在一起完成程式設計，用拖曳、組合的方式取代打字，免除指令輸入錯誤的困擾，更簡單、更直覺，非常容易上手，；程式區塊中的指令依照顏色可區分為動作、外觀、聲音、畫筆、控制、偵測、運算及變數 8 種類型，如圖 4 所示，動作指令與角色移動、旋轉及位置相關，外觀指令與角色造型、說話、顏色、大小及圖層相關，聲音指令與音樂、音效及節奏相關，畫筆指令與筆跡、畫筆顏色及尺寸相關，控制指令與判斷、廣播、迴圈及條件句相關，偵測指令與感應、傳回值及接觸相關，運算指令與四則運算、布林運算及大小比較相關，變數指令與建立變數及陣列相關，共計超過 90 個指令供使用者使用。



圖 2 Scratch 介面



圖 3 Scratch 程式



圖 4 Scratch 指令

一個 Scratch 程式包含一個固定的舞台及許多可移動的角色，每個物件擁有自己的聲音、造型及程式碼，這種結構使得角色的輸出及轉換更加容易。程式碼編輯是將程式區塊中的指令拖曳至程式區並進行組合，根據需要可修改格子內的參數值，指令被設計成只有符合語法才能相互組合，可避免語法錯誤的困擾，而不同的資料型態有不同的形狀，可避免資料型態配置錯誤。已組合的程式，只要拖曳最上方一個，即可全部一起移動，只要在最外層的控制指令點擊二下，程式即開始執行並在舞台中直接顯示程式執行結果，如果有多個動作需要判斷或執行，如按下綠旗、鍵盤或滑鼠，可利用帽型控制指令來引發程式，同時執行多段程式，解決與時間運作相關的事件，程式執行時，可同步修改程式區指令，使用者有任何新構想時，可隨時立即新增以及反覆進行試驗。Scratch 同時展示舞台、角色及程式碼的設計，方便使用者了解電腦如何運作邏輯程式，可以加強學習者對程式設計流程的認知，提升高層次思考能力，Scratch 指令簡單，但擁有創造複雜程式的能力，具有多樣性功能，以程式設計者的觀點來解題，利用日常用語，程式設計者容易將自己想像成舞台中的角色來完成設計工作，整體而言，是一種容易學習和掌握的程式語言，適合程式初學者入門，研發者認為 Scratch 適用於八歲以上的兒童學習數學以及計算能力，藉由這些學習，八歲以上的兒童也可輕鬆學會寫程式。

Scratch 支援多國語言，1.3 版後能支援中文、日文等多位元文字，並可在網路上分享自己的作品，具有強大的繪圖能力，以簡單的動作及畫筆指令，即可繪製許多複雜圖形，提供使用者自訂程序，可使用自訂程序來完成程式碼模組建構，可執行呼叫，能簡化許多複雜的程式設計工作，可用來創造互動式故事、動畫、遊戲、音樂和藝術。Scratch 幫助培養學生創造性思考，在學習程式設計過程中，可與數學幾何、算術及統計

概念相互配合，探索解題的要領與技巧，學生可學習如迴圈、條件句、變數、資料類型、事件及程序等重要的程式概念，從小學至大學，Scratch 已用來對很多不同年齡的學生介紹程式概念，作為學生學習高階程式語言的過渡工具。因此本研究以修習 C++ 程式設計的大一學生為研究對象，針對低成就組(期中考成績在全班四分之一以下)的 12 位學生，選擇以 Scratch 對全班進行補救教學。

### (三) 研究對象與實驗設計

本研究以國立臺南大學數位學習科技學系修習 C++ 程式設計的一年級學生 45 人為研究對象，其中男生 32 人，女生 13 人，以學習焦慮量表以及有趣性與愉悅性問卷，探討學習成就、學習焦慮與程式語言之間的關係，有效問卷樣本 45 筆。首先在期中考後使用學習焦慮量表施測，並收集學生對 C++ 的有趣性與愉悅性的問卷反應；其次針對低成就組(期中考成績在全班四分之一以下)的 12 位學生，選擇以 Scratch 對全班學生進行補救教學；補救教學後再次以學習焦慮量表施測，並收集學生在學過 Scratch 後，對程式設計概念的有趣性與愉悅性的問卷反應；研究分析學生期中考(補救教學前)及期末考(補救教學後)之成績；最後對低成就組的 12 位學生進行非正式訪談，了解學生學習 Scratch 的感受與經驗，並分析補救教學中的 Scratch 作品，以了解影響學習成就的要素。

## 四、實驗結果與討論

### (一) Scratch 補救教學後對 C++ 學習焦慮之影響

為了解以 Scratch 進行補救教學後對學生 C++ 的學習焦慮是否有顯著影響，使用學習焦慮量表，在期中考後實施前測，以 Scratch 進行補救教學後實施後測。求出前、後測學習焦慮量表分數後，分別對全班及低成就組學生進行成對樣本 t 檢定，結果如表 1 所示，補救教學後，全班

t = 5.0789，顯著性 = .0000 < .001，低成就組 t = 4.3551，顯著性 = .0011 < .01，皆達顯著水準，而由平均數分析得知，全班及低成就組學生後測表現皆優於前測，全班平均減少 0.4520 分，低成就組平均減少 0.7125 分，這個結果說明以 Scratch 進行補救教學後對學生 C++ 的學習焦慮確實有顯著影響，明顯降低學生學習焦慮。以 Scratch 對學生進行補救教學後，表示害怕學習 C++ 程式設計的學生，全班由 48.89% 降低為 20.00%；低成就組由 66.67% 降低為 33.33%；對學習 C++ 程式設計感到擔心的學生，全班由 46.67% 降低為 22.22%，低成就組由 50.00% 降低為 25.00%；想到學習 C++ 程式設計就心情沉重的學生，全班由 31.11% 降低為 17.78%，低成就組由 50.00% 降低為 16.67%；整體學習焦慮全班由平均 41.98% 降低為 23.21%，低成就組由平均 59.26% 降低為 27.78%，低成就組顯著性雖不如全班，但學習焦慮的降低程度優於全班。

表 1 Scratch 補救教學前後之 C++ 學習焦慮成對樣本 t 檢定

	個數	補救教學前		補救教學後		t	自由 度	顯著性 (雙尾)
		平均數	標準差	平均數	標準差			
全班	45	3.366	0.858	2.914	0.612	5.079	44	0.000 ***
低成就組	12	3.740	0.677	3.028	0.515	4.355	11	0.001 **

\*p < .05, \*\*p < .01, \*\*\*p < .001

## (二) Scratch 補救教學後對 C++ 有趣性與愉悅性之影響

為了解以 Scratch 進行補救教學後對學生 C++ 的有趣性與愉悅性是否有顯著影響，使用有趣性與愉悅性問卷，同樣在期中考後實施前測，以 Scratch 進行補救教學後實施後測。求出前、後測有趣性與愉悅性問卷分數後，分別對全班及低成就組學生進行成對樣本 t 檢定，結果如表 2 所示，補救教學後，低成就組在有趣性方面，t = -2.4452，顯著性 = .0325 < .05，達顯著水準，全

班在愉悅性方面 t = -2.0648，顯著性 = .0449 < .05，達顯著水準；全班在有趣性方面，低成就組在愉悅性方面，雖皆未達顯著水準，但由平均數分析得知，全班及低成就組學生後測表現皆優於前測，這個結果說明以 Scratch 進行補救教學後對學生 C++ 的有趣性與愉悅性確實有影響，雖不明顯，但確實可提高學生有趣性與愉悅性。以 Scratch 對學生進行補救教學後，整體有趣性全班提高 3.81%，低成就組提高 15.48%，整體愉悅性全班提高 5.93%，低成就組提高 19.44%，低成就組有趣性與愉悅性提高程度皆優於全班。

表 2 Scratch 補救教學前後之 C++ 有趣性與愉悅性成對樣本 t 檢定

	個數	補救教學前		補救教學後		t	自由 度	顯著性 (雙尾)	
		平均數	標準差	平均數	標準差				
有趣性	全班	45	2.976	0.591	3.093	0.471	-1.475	44	0.147
	低成就組	12	2.645	0.373	3.097	0.569	-2.445	11	0.033 *
愉悅性	全班	45	2.926	0.838	3.163	0.809	-2.065	44	0.045 *
	低成就組	12	2.694	0.731	3.111	0.891	-1.585	11	0.141

\*p < .05, \*\*p < .01, \*\*\*p < .001

## (三) 降低學習焦慮後對 C++ 成績之影響

為了解以 Scratch 進行補救教學後對學生 C++ 的成績是否有顯著影響，分析學生期中考(補救教學前)及期末考(補救教學後)之成績，分別對全班及低成就組學生進行成對樣本 t 檢定，結果如表 3 所示，補救教學後，低成就組 t = -5.2675，顯著性 = .0003 < .001，達顯著水準；而全班雖未達顯著水準，由平均數分析可得知，全班期末考成績優於期中考成績，這個結果說明以 Scratch 進行補救教學後對學生 C++ 的成績確實有影響，確實可提高學生成績，其中以低成就組學生的成績進步最為明顯，平均進步 33 分。檢視學生成績的進退步人數及比例，如表 4 所示，補救教學後，就全班學生而言，有半數以上的學生成績進步或維持不變；就低成就組學生而言，有八

成以上的學生成績進步或維持不變，以折線圖的方式呈現低成就組學生補救教學前後之成績變化，如圖 5 所示，雖有 2 位學生補救教學前後之成績皆為 0 分，但有 10 位學生成績皆有所進步，其中最少為 15 分，最多達 60 分，更有 7 位學生由不及格進步至及格，由此可推論以 Scratch 進行補救教學後對低成就組大部份學生 C++ 的成績確實是有幫助的。

表 3 Scratch 補救教學前後之 C++ 成績成對樣本 t 檢定

個數	補救教學前		補救教學後		t	自由度	顯著性(雙尾)	
	平均數	標準差	平均數	標準差				
全班	45	53.889	32.641	60.444	24.400	-1.405	44	0.167
低成就組	12	14.167	15.523	46.667	23.570	-5.268	11	0.000 ***

\*p < .05, \*\*p < .01, \*\*\*p < .001

表 4 全班及低成就組 C++ 成績進退步人數統計表

	全班	低成就組
全部人數	45 (100.00%)	12 (100.00%)
成績進步	25 (55.56%)	10 (83.33%)
成績退步	17 (37.78%)	0 (0.00%)
成績不變	3 (6.67%)	2 (16.67%)

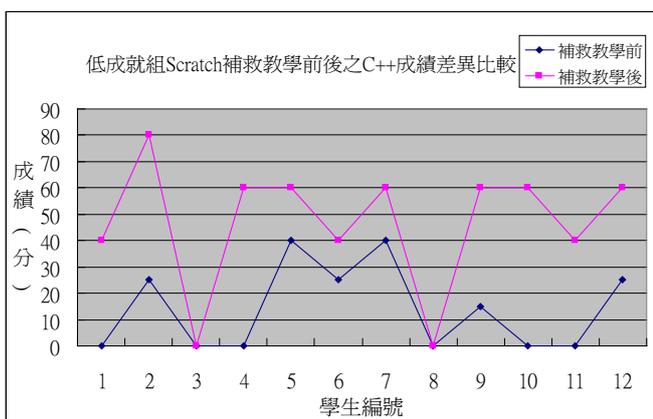


圖 5 低成就組補救教學前後之成績比較

#### (四) Scratch 作品分析

本研究針對低成就組(期中考成績在全班四分之一以下)的 12 位學生進行非正式訪談，並分

析補救教學中的 Scratch 作品，為了解學生使用 Scratch 學習到哪些程式概念，分析這 12 個作品所使用的程式指令類型，其中男生 8 人，女生 4 人。這 12 個作品的程式指令皆使用循序執行，學生平均撰寫 6 個 Script，使用 60 個指令，最少的撰寫 1 個 Script，使用 8 個指令，最多的撰寫 12 個 Script，使用 169 個指令，除了將程式指令分為動作、外觀、聲音、畫筆、控制、偵測、運算及變數八種類型，尚依照程式概念分為使用者互動(鍵盤或滑鼠輸入)、溝通和同步(廣播和接收)、迴圈、條件句、邏輯運算、亂數及變數七種類型來進行分析。如圖 6 所示，觀察程式指令，由於動畫與遊戲的普及，不難發現學生使用外觀、控制的比率最高達 33% 以上；動作大約佔 15%；聲音大約佔 9%；偵測、運算及變數的比例則偏低，平均佔不到 2%。在程式概念中，學生運用迴圈的比例最高達 46% 以上；變數大約佔 14%；條件句大約佔 10%，學生撰寫較複雜程式時，不免使用迴圈及條件句，此時須利用變數；邏輯運算及亂數的比例仍偏低，平均佔 5%，Scratch 雖容易使用，低成就組學生對學習邏輯運算仍是有較大的困難及憂慮；另外有學生運用使用者互動以及溝通和同步的概念，雖然平均只佔 9%，但令人鼓舞，這是 Scratch 最複雜的程式概念之一，提供程序平行處理的程式概念，在撰寫複雜程式時提供一個關鍵的需要。

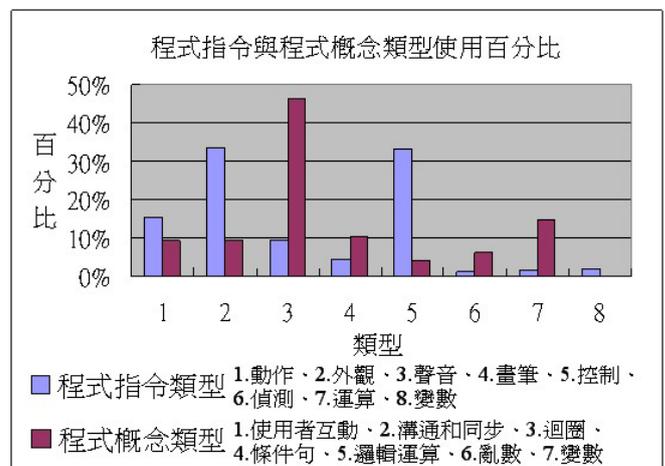


圖 6 程式指令與程式概念類型使用百分比

## 五、結論

本研究以修習 C++ 程式設計的大一學生為研究對象，探討學習成就、學習焦慮與程式語言之間的關係。在期中考及補救教學後皆使用學習焦慮量表施測，並分別收集學生對 C++ 的有趣性與愉悅性的問卷反應；研究選擇以 Scratch 對全班進行補救教學，針對低成就組(期中考成績在全班四分之一以下)的 12 位學生進行非正式訪談，並分析補救教學中的 Scratch 作品。結果顯示學習成就低的學生學習焦慮較高，尤其以低成就組(期中考成績在全班四分之一以下)的 12 位學生最為明顯，以 Scratch 進行補救教學後，對學生 C++ 的學習焦慮及有趣性與愉悅性都確實有顯著影響，明顯降低學生學習焦慮，提高學生有趣性與愉悅性，其中低成就組學生的學習焦慮降低程度及有趣性與愉悅性提高程度都較高，分析學生補救教學前後之成績也以這 12 位學生的成績進步最為明顯。大部分學生接觸 Scratch 進程式編碼後，都覺得學習 Scratch 非常有趣且新奇，程式設計並不是那樣令人畏懼。整體而言，研究發現使用 Scratch 能在悅趣的氣氛下進行有效的補救教學，並提高學生成績，值得程式設計教師參考採用。

## 六、致謝

本研究承蒙國科會專題研究計畫贊助，計畫編號 NSC 97-2628-S-024-001-MY3、NSC 98-2631-S-024-001、NSC 98-2511-S-024-008。

## 七、參考文獻

- [1] 吳明隆 (1995)。國民小學學生電腦態度及其相關因素之研究。《教育研究資訊》，36，71-86。
- [2] 杜正治(1993)。補救教學的實施。載於李永吟(主編)，學習輔導：學習心理學的應用。台北市：心理出版社。
- [3] 林振輝 (2003)。應用 Bottom-Up 法進程式教學。論文發表於第 11 屆國際電腦輔助教學研討會(ICCAI2003)。國立台灣師範大學，台北市。
- [4] 林震城 (1997)。兩岸大學生電腦態度與電腦素養之比較研究。國立中央大學資訊管理研究所碩士論文，未出版，桃園縣。
- [5] 邱上真(1992)。學習障礙兒童的教育評量—認知取向。《特殊教育季刊》，43，1-6。
- [6] 高廣孚(1988)。教學原理。台北市：五南出版社。
- [7] 張春興 (1999)。教育心理學：三化取向的理論與實踐。台北市：東華書局。
- [8] 張新仁(1995)。教學原理與策略。載於王家通(主編)，教育導論。台北市：五南出版社。
- [9] 張新仁(2000)。補救教學面面觀。論文發表於義務教育階段補救教學系統研究與實務研討會。國立高雄師範大學特殊教育中心，高雄市。
- [10] 張新仁(2001)。實施補救教學之課程與教學設計。《國立高雄師範大學教育學系教育學刊》，17，85-106。
- [11] 張新仁、邱上真、李素慧(1999)。國中學習困難學生之補救教學方案研究。行政院國家科學委員會專題研究成果報告(NSC 88-2614-H-017-011)。
- [12] 黃淑苓(1999)。補救教學之設計與實施。論文發表於學習落後學生的補救教學與輔導研討會。國立中興大學教育學程中心，台中市。
- [13] 董少桓、林宗德、沈維倫、錢傳明和沈勇嘉 (2005)。支援教師分身的智慧型程式語言教學平台。論文發表於 2005 年全國計算機會議(NCS2005)。崑山科技大學，台南縣。
- [14] Brusilovsky, P., & Spring, M. (2004). Adaptive, engaging, and explanatory visualization in a C programming course. In L.

- Cantoni & C. McLoughlin (Eds.), *Proceedings of World Conference on Educational Media, Hypermedia, and Telecommunications 2004* (pp. 1264-1271). Chesapeake: VA: AACE.
- [15] Govender, I., & Grayson, D. (2006). Learning to program and learning to teach programming: A closer look. In E. Pearson & P. Bohman (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006* (pp. 1687-1693). Chesapeake: VA: AACE.
- [16] Julie, M. P., & Ruby, E. (2006). Supplemental instruction in developmental mathematics. *The Community College Enterprise*, 12, 21-38.
- [17] Kirsner, K. (1998). *Implicit and explicit mental processes*. Mahwah NJ: LEA.
- [18] Maurer, M. M., & Simonson, M. R. (1984). *Development and validation of a Measure of computer anxiety*: Paper presented at the Annual Meeting of the Association for Educational Communications and Technology.(ERIC Document Reproduction Service No.ED243428).
- [19] Shanmugasundaram, V., Juell, P., Groesbeck, G., & Makosky, M. (2006). Evaluation of Alice World as an Introductory Programming Language. In E. Pearson & P. Bohman (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006* (pp. 1976-1982). Chesapeake: VA: AACE.
- [20] Venkatesh, V. (2000). Determinants of Perceived Ease of Use: Integrating Control, Intrinsic Motivation, and Emotion into the Technology Acceptance Model. *Information Systems Research*, 11(4), 342-365.
- [21] Winslow, L. E. (1996). Programming pedagogy: A psychological overview. *SIGCSE Bulletin*, 28, 17-22.
- [22] Yi, M. Y., & Hwang, Y. (2003). Predicting the use of web-based information systems: self-efficacy, enjoyment, learning goal orientation, and the technology acceptance model. *International Journal of Human-Computer Studies*, 59(4), 431-449.
- [23] Young, D. J. (1991). Creating a low-anxiety classroom environment: What does language anxiety research suggest? *The Modern Language Journal*, 75, 423-426.

## 八、附錄

### 學習 C++ 程式設計調查問卷

親愛的同學您好：

這是一份學術性研究問卷，主要目的是希望了解同學學習 C++ 程式設計的經驗。希望能藉分析所得之原始資料，了解學生在學習 C++ 程式設計的焦慮、態度及滿意度。有關您學習 C++ 程式設計之量表，每個問題並無標準答案或對與錯，請您依直覺，就同意與否程度，在適當  內打<sup>✓</sup>。本問卷採記名的方式，但不計分，問卷內容僅供學術分析用，不對外公開，敬請放心填答。感謝各位同學的幫忙，您的參與對我們的研究將助益繁多。祝 身體健康、學業進步！

國立臺南大學數位學習科技學系 張智凱老師

一、個人基本資料	
1. 學 號：_____	2. 姓 名：_____
3. 系 級：_____	4. 性 別： <input type="checkbox"/> 男 <input type="checkbox"/> 女
二、學習焦慮	
1. 我一點也不害怕學習 C++ 程式設計。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
2. 學習 C++ 程式設計使我覺得緊張。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
3. 當其他人談論學習 C++ 程式設計時，我並不會感覺到威脅。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
4. 學習 C++ 程式設計並不會使我覺得困擾。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
5. 學習 C++ 程式設計令我覺得很不自在。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
6. 我覺得學習 C++ 程式設計很輕鬆。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
7. 想到要學習 C++ 程式設計，我的心情 便很沉重。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
8. 學習 C++ 程式設計令我覺得很自在。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
9. 想到要學習 C++ 程式設計，我便有一點擔心。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
三、有趣性	
1. 我會自動自發地學習 C++ 程式設計。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
2. 我覺得學習 C++ 程式設計很無趣。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
3. 我很容易適應學習 C++ 程式設計。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
4. 學習 C++ 程式設計使我更有創造力。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
5. 我不是很認真地學習 C++ 程式設計。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
6. 學習 C++ 程式設計時，我是模仿的。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
7. 學習 C++ 程式設計時，我沒有發明能力。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
四、愉悅性	
1. 學習 C++ 程式設計是一件令我感到愉快的事。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
2. 學習 C++ 程式設計的實際過程令我感到愉快。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
3. 學習 C++ 程式設計使我獲得樂趣。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意

## 學習程式設計概念調查問卷

親愛的同學您好：

這是一份學術性研究問卷，主要目的是希望了解同學學習程式設計概念的經驗。希望能藉分析所得之原始資料，了解學生在學過 Scratch 之後，對學習程式設計概念的學習焦慮、遊戲性及愉悅性。有關您學習程式設計概念之量表，每個問題並無標準答案或對與錯，請您依直覺，就同意與否程度，在適當□內打√。本問卷採記名的方式，但不計分，問卷內容僅供學術分析用，不對外公開，敬請放心填答。感謝各位同學的幫忙，您的參與對我們的研究將助益繁多。祝 身體健康、學業進步！

國立臺南大學數位學習科技學系 張智凱老師

一、 個人基本資料	
1. 學 號：_____	2. 姓 名：_____
3. 系 級：_____	4. 性 別： <input type="checkbox"/> 男 <input type="checkbox"/> 女
二、 學習焦慮	
1. 學過Scratch後，我一點也不害怕學習程式設計概念。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
2. 學過Scratch後，學習程式設計概念使我覺得緊張。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
3. 學過Scratch後，當其他人談論學習程式設計概念時，我並不會感覺到威脅。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
4. 學過Scratch後，學習程式設計概念並不會使我覺得困擾。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
5. 學過Scratch後，學習程式設計概念令我覺得很不自在。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
6. 學過Scratch後，我覺得學習程式設計概念很輕鬆。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
7. 學過Scratch後，想到學習程式設計概念，我的心情便很沉重。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
8. 學過Scratch後，學習程式設計概念令我覺得很自在。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
9. 學過Scratch後，想到要學習程式設計概念，我便有一點擔心。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
三、 有趣性	
1. 學過Scratch後，我會自動自發地學習程式設計概念。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
2. 學過Scratch後，我覺得學習程式設計概念很無趣。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
3. 學過Scratch後，我很容易適應學習程式設計概念。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
4. 學過Scratch後，學習程式設計概念使我更有創造力。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
5. 學過Scratch後，我不是很認真地學習程式設計概念。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
6. 學過Scratch後，學習程式設計概念時，我是模仿的。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
7. 學過Scratch後，學習程式設計概念時，我沒有發明能力。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
四、 愉悅性	
1. 學過Scratch後，學習程式設計概念是令我感到愉快的事。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
2. 學過Scratch後，學習程式設計概念的實際過程令我感到愉快。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意
3. 學過Scratch後，學習程式設計概念使我獲得樂趣。	<input type="checkbox"/> 非常不同意 <input type="checkbox"/> 不同意 <input type="checkbox"/> 普通 <input type="checkbox"/> 同意 <input type="checkbox"/> 非常同意