

Alice 程式設計環境中配對與個別之學習成效比較

賴錦緣

國立臺中科技大學通識教育中心副教授

摘要

程式設計可培養學生問題解決和邏輯思考能力，但對初學者而言是相當困難的，因此本研究以 3D 動畫程式設計環境 Alice 來促進學生的學習成效。研究採準實驗設計，探討學生以 Alice 進行程式設計學習時，運用不同學習方式（配對學習，即兩人一組方式與個別學習）在動畫成果與學習態度的差異。研究參與者為中部五專一年級之兩班學生，其中一班 51 人採配對學習，另一班 50 人以個別學習方式。研究結果發現，配對組的學習成效優於個別學習組，對自己製作的 Alice 動畫作品感到驕傲的比例亦較高；但個別學習組學生認為 Alice 有趣的比例卻高於配對組。研究也發現學生學習 Alice 時遇到一些問題，包括英文介面障礙、電腦有時容易當機、不易做出精緻動畫、及信心不足等。建議未來教學可以延長授課時間，並先讓學生個別操作軟體以對軟體熟悉與建立信心之後，再讓學生進行程式語言的配對學習。

關鍵詞：Alice、配對程式設計、程式設計

Comparison of Learning Effects between Pair Programming and Individual Learning in the Alice Programming Environment

Chin-Yuan Lai

Associate Professor, Center for General Education, National Taichung University of Science and Technology

Abstract

Programming can cultivate students' abilities of problem solving and logical thinking. However, it is very difficult for beginners. The purpose of this study was to use an innovative 3D Alice programming environment to improve students' learning. A quasi-experimental research design was adopted to compare the effects of using pair- and individual-programming strategies on students' learning performances and learning attitudes toward Alice programming. The participants were two classes of first-year students in a junior college in central Taiwan. One class of 51 students used pair programming to design animation. The other class of 50 students used the traditional individual learning method. The results indicated that students' learning performances in the pair-programming group were higher than in the individual learning group and they had more pride on their animation work. However, the students in the individual learning group showed more interest in using Alice to design animation. The study also found that students had problems with learning Alice such as English interface barriers, computers sometimes crashing, sophisticated animations not easy to make, as well as students lacking in confidence. Future research may extend teaching time, allowing students to first operate Alice individually with confidence and familiarity, and then conduct pair programming.

Keywords: Alice, Pair programming, Programming

壹、緒論

一、研究動機

程式設計是電腦科學教學中很重要的部分，其教學目標是要讓學生根據程式語言的語法、語言結構與設計技巧來解決問題，進而提升其程式設計技巧與邏輯思考能力。相關研究指出，學習程式設計的學生在問題遷移與程序理解的能力有較佳的表現，而且除了習得運用程式解決問題之外，還可進一步將問題解決的技巧運用到其他領域 (Linn, 1985; Mayer, Dyck, & Vilberg, 1986)。然而，雖然程式設計的學習相當重要，但程式設計涉及到許多認知的技能，初學者在尋求問題解決辦法，通常要面對複雜的語法與指令，這些抽象的結構與陌生的名詞，使得學習程式設計被認為是一項困難且複雜的工作 (尹玫君, 1991; Kelleher & Pausch, 2005)。因此，如要改善學習程式設計的困境，應選擇適於學習的程式語言、採取彈性的教學方法、避免以評量為目的的教學、以及改善傳統沒有顧及學生個別差異的教學方法，以使學生得到更多回饋並增加其學習興趣 (Jenkins, 2002; Anabela & Mendes, 2007)。

Alice 是一套創新的 3D 程式設計環境，使用者可以透過說故事的方式進行動畫設計、參與互動式的遊戲、以及將影片上傳到網路與他人分享。它是由美國賓州卡內基美隆大學 (Carnegie Mellon University) 所發展出來，用於協助初學者以物件導向 (object-oriented) 觀念來學習程式設計基本概念之免費教學工具。相關研究顯示，Alice 不僅可以增進學習者問題解決和演算法思考的能力，也能有效改善學生的學習態度、以及促進學生的合作學習 (Bishop-Clark, Courte, & Howard, 2006; Howard, Evans, Courte, & Bishop-Clark, 2009)。國內將 Alice 用於高中電腦課程之程式語言教學，並與傳統程式語言 C++ 相較，結果發現 Alice 更能有效率地在有限時間內進程式概念的教學 (王鼎中等, 2009)。

讓學生以兩人一組的方式來進程式設計活動，稱為配對程式設計 (pair programming)，在軟體工程界稱為極限編程 (Extreme Programming, XP)，是近年頗受歡迎的程式開發模式 (Beck, 1999)。其特色是兩位程式設計師同時共用一台電腦進行工作，一人為操作者 (driver)，負責撰寫程式碼；另一人為觀察者/航海家 (observer/navigator)，負責檢視操作者的工作 (Williams & Kessler, 2002)。研究顯示使用配對程式設計進行教學有許多優點，包括：(1) 撰寫的程式品質更高；(2) 減少程式完成時間；(3) 幫助理解程式結構；(4) 增加程式學習的樂趣；(5) 改善學生修課的完成率；(6) 增強學生學習成就 (Preston, 2005)。因此許多學者將配對程式設計引入學校的教學，讓學生以兩人一組的方式進程式設計活動，以取代傳統教學，讓學生獨自一人撰寫程式的個別學習方式 (Braughr, Eby, & Wahls, 2008; McDowell, Werner, Bullock, & Fernald, 2002; Simon & Hanks, 2007; Williams & Kessler, 2002)。

然而，在 Alice 動畫設計環境下，相關研究對於採取配對程式設計或個別學習的教學成效卻有不一致的結果。Bishop-Clark 等 (2006) 與 Howard 等 (2009) 之研究，比較獨自學習 Alice 與兩人一起學習 Alice 的差異，發現兩人一組的配對學習方式，學生學習興趣更高，對演算法的瞭解及其與 Alice 程式設計的關係，也有顯著成效。然而，

Kelleher (2006) 的研究中，卻發現兩人一組的學習方式其學習效果並不佳，因為兩人常常在協商故事情節時遇到困難，因而導致對話中止。因此，本研究擬針對這一部份進行釐清與探討，瞭解 Alice 配對學習和個別學習對學生學習表現和態度的影響。

二、研究目的

本研究的研究目的包括：

1. 在 Alice 程式設計環境下，比較配對與個別組學生程式設計學習表現的差異。
2. 在 Alice 程式設計環境下，比較配對與個別組學生程式設計學習態度的差異。

貳、文獻探討

一、Alice 程式設計環境

Alice 是一套 3D 動畫程式開發環境，它是由美國卡內基麥隆大學 (Carnegie Mellon University) 所發展的免費教學軟體，目標是幫助青少年學習電腦程式，包括學習程式的基本概念、學習問題解決策略、以及探索物件導向程式設計概念等。開發團隊希望這套軟體能夠幫助青少年實現學習程式設計的夢想，讓從未接觸過程式設計的人，不論中學生或大學生，都能夠輕鬆地學習基礎電腦科學與程式設計。

Alice 提供情境式的環境，可幫助使用者建立程式執行的心智模型，它藉由 3D 動畫來動態地顯示程式執行的歷程與結果，使用者可以清楚地理解他們的程式所產生的行為是否正確。如圖 1，Alice 以圖形介面來建立與編輯程式碼，運用拖曳指令方式來控制 3D 動畫，使用者可專注於問題的解決，無需自行打字，因此可有效減少程式初學者常犯的語法錯誤問題。然而，與其他圖形介面的教學軟體不同的是，Alice 並非只是一個玩具，它還包涵了選擇、計次迴圈、條件迴圈、變數、陣列、副程式、平行處理等程式設計概念與功能。此外，Alice 提供了語法互換的功能，可將程式碼轉換成 C++ 或 Java 語法，有助於引導學生進一步學習 C++ 或 Java 語言。



圖 1 Alice 使用圖形介面來撰寫程式碼

二、Alice 教學相關研究

Alice是一個非常容易學習和使用的3D發展環境，允許使用者透過建構虛擬世界與動畫物件之過程中來學習程式設計的概念。Copper, Dann與Pausch（2000a）的研究中，觀察到Alice有幾項特徵適合程式設計初學者使用：

1. 立即性的回饋：大部分程式語言在執行前都必須編譯（compile）程式，但在Alice環境中可以馬上看到程式執行的結果。這種高度視覺化的回饋可以讓學生將程式片段與動畫的動作進行關聯，有助於程式功能結構的理解。Copper等人認為這種視覺化的效果就像機器人模擬軟體Karel, The Robot（一個可程式化的軟體機器人，它可以依照使用者撰寫的指令在二維的格子中移動，使用者在螢幕即可以看到Karel執行指令的情況）一樣，可以讓學生建立他們自己獨特的世界。
2. 有趣：Copper他們觀察發現學生喜愛3D動畫世界的建構，學生通常花了比預期還要多很多的時間來建立更為逼真的動畫序列。
3. 狀態：在Alice中撰寫程式，通常不需要使用傳統的變數方式，即可看到物件狀態被改變後的結果，因此可以讓學生花較多時間專注於不同程式結構的理解。
4. 合作學習：學生以小組方式在一起設計之動畫專題比個人還要優秀，組員合作無間讓物件的移動更逼真，因此學生對他們的作品感到驕傲，甚至用email寄給朋友或上傳到網路與人分享。

5. 自然語言：Alice程式語言類似英文，所有的指令皆有物件（名詞）、動作（動詞）、以及非必要的參數（形容詞和副詞）。例如：「`Bunny.Move(Forward, 1, Duration=3)`」這個指令，表示讓小兔子往前移動一個單位，而且必須在三秒內完成。
6. 物件觀念：Alice有強烈的物件導向程式設計的特性。雖然在Alice並沒有教導學生物件的觀念，但學生在動畫製作過程中，已經接觸到資料隱藏、繼承、物件和方法等概念。例如，在資料隱藏方面，學生只要知道如何設定讓小兔子往前移動一個單位，並不需要知道小兔子如何移動的。

相關研究顯示，Alice不僅可以改善學生對程式設計學習的態度，也能增進學習者在問題解決和演算法思考的能力。在增進程式設計相關概念方面，Cooper等（2000a）發現Alice讓學生建構動畫的過程中，即提供了一個問題解決以及引導演算法思考的環境。在改善學生學習態度方面，Cooper, Dann,與Pausch（2000b）指出，原本程度就高的學生，因為Alice更增進了學習動力，樂意主動投入更多時間修改程式碼，以觀察程式碼對動畫的影響；學生對自己發展演算法設計程式的能力充滿信心，對物件、方法、程式結構展現直覺能力。Bishop-Clark等（2006）也發現Alice使得學生對程式設計的態度轉為正面，尤其對於之前被認定在這個課程低學習成就的學生更是明顯。

三、配對程式設計

以下描述配對程式設計的學習原理、配對行為、以及配對與個別程式設計比較的相關研究。

Chong等人（2005）認為配對程式設計能對學習者產生幫助的原因，在於程式設計的過程中必須不斷地討論與除錯，此兩個步驟皆須與夥伴交換意見與分享知識，學生藉由互動的過程，不斷擴大知識的範疇，這是學生個別進行程式設計所做不到的。Williams與Upchurch（2001）指出學生在進行配對程式設計時，小組成員藉由溝通來進行問題分析及分享解法以完成任務，因此小組成員間不斷地產生知識，進而達到較好學習的效果。從心理學的角度來解釋，小組成員間因為擁有不同的經驗、知識、看法，透過互動而產生的分散性認知（distributed cognition），使得小組能得到更多元的解法，進而得到較好的答案（Flor, 1998）。

Williams及Kessler（2002）指出小組在進行配對程式的過程中，會產生七種有利於學習的配對行為，包括：配對壓力、配對協商、配對勇氣、配對檢查、配對除錯、配對學習與配對信任。Cao及Xu（2005）發現在進行配對程式設計的過程中，小組成員的互動行為可分成詢問意見、要求解釋、提出意見、及評論與總結現況等四種。其研究更進一步分析三種不同能力的配對小組發現：(1)小組成員能力較高的組別，其進行配對活動時，皆會出現此四種行為；(2)小組成員能力皆為中等的組別，其四種行為雖會發生，但在詢問意見時通常為表面的互動，較少進一步討論，且有時會提供錯誤的解釋；(3)小組成員的能力為一高一低者，他們只有在一開始配對時會提供意見給對方，且會解釋程式的細節給自己的夥伴，至於批評同伴與總結現況則沒有出現。

Williams等人（2002）比較CS1（Computer Science 1）課程中使用配對程式設計或個別程式設計進行教學習時學生程式設計表現與態度之差異，發現：

1. 參與配對程式設計學生的修課完成率為68%，而進行個別程式設計的修課完成率為45%，兩者的比例達到統計上的顯著差異。
2. 實施配對程式設計的學生，其成就測驗成績與個別組學生的成績在統計上並無顯著差異，其可能原因為：(1)配對程式設計使原本因成績差而想停修課程的學生繼續修完課程，因此降低整體平均；(2)配對程式設計並無訓練學生回答簡答題與選擇題的能力，學生可能因答題技巧的因素，使得平均分數與個別組無顯著差異。
3. 評估學生所製作的三次專題，前兩次實施配對程式設計的組別，其所設計的專題皆優於個別組學生，但第三次的專題則為個別組學生較好，其原因可能為個別組中表現較差的學生已經停止修課造成的影響。
4. 針對自信心、動機、態度、性別與課程有用性等五個方面進行調查發現，配對組與個別組的調查結果並無顯著差異，因此無法證實配對程式設計能帶給學生正面的態度。

另外，黃建庭（2003）的研究指出，實施配對程式設計的學習成就並未高於個別程式設計，但學生認為配對程式設計對學習有幫助，如：可增加解題想法、減少錯誤、增進同學間的關係等。此外，其研究也發現學生對於實施配對程式設計的意願不高，且採用個人偏好的配對方式會造成學生容易討論與課堂無關的話題，使得教室管理不易。可見配對程式設計在教學的應用上，仍有許多值得探討的問題，其與個別學習成效之差異，也受到各種因素的影響，因此各研究結果並不一致。

參、研究方法

本研究主要探討在Alice程式設計環境下，配對學習（配對組）和個別學習（個別組）的學生在學習成效的差異。研究採取準實驗設計，以下針對本研究的參與者、教學活動及資料蒐集與分析進行說明。

一、研究參與者

本研究以方便取樣，選擇中部某科技大學五專一年級學生102學年度修習「計算機概論」（該課程為2學分，每週兩小時）的二班學生為參與者，全部皆為女性。其中一班為配對組（51人），另一班為個別組（50人），參與的學生都是第一次接觸Alice軟體，入學時為常態分班，如表1，兩班於該課程的期中考分數並未達顯著差異。

表 1 兩班學生期中考分數統計

	人數	<i>M</i>	<i>SD</i>	<i>t</i>
配對組	51	76.2	6.6	1.31
個別組	50	78.0	6.7	

二、教學活動

不管配對學習組或個別組，授課時間皆為五週，每週兩節課，每節50分鐘。教材內容參考高慧君的Alice網站（<http://lib.nksh.tp.edu.tw/jean/lessons/index.php?page=alice>）。內容包括：Alice開發環境、Alice場景布置、Alice基本概念（如：物件、屬性、方法）、

以及程式設計概念（如循序結構Do in order、平行結構Do together、重覆結構Loop和While、選擇結構If/Else）等。前四週的課堂以「龜兔賽跑」為例子，採每週逐步擴充功能方式，以進行這些概念與操作的練習，最後一週則要求學生繳交一份自創的Alice故事動畫作品。

三、資料蒐集與分析

本研究蒐集與分析的資料包括學生的Alice作品和態度問卷，茲說明如下。

(一) Alice 作品

本研究蒐集學生所設計的Alice動畫作品進行評分，以比較兩班學生進程式設計學習的成效。其中配對組51人有25個作品（其中24組為2人一組，只有一組是3人）；個別組50人有50個作品。評分的標準如表2，評分項目總共有五項，包括：場景安排、故事主題、動畫流暢性、多媒體與創意；每一個項目的評分則分為優秀（4分）、熟練（3分）、發展（2分）與初期（1分）四個等級。為提高評分者信度，此評分由作者與另一位電腦教師共同進行，並將其評分進行分析者間信度考驗，結果顯示五個向度的分析者間信度分別為0.85、0.91、0.88、0.95與0.78。評分結果出爐後，最後以描述性統計方法計算每班各分項分數的平均數與標準差，再以獨立樣本t 檢定檢測兩組的分數是否有差異性存在。

表2 Alice動畫作品評分準則

評分項目/等級	4分（優秀）	3分（熟練）	2分（發展）	1分（初期）
1. 場景安排	布置良好且具複雜性	布置良好	布置 OK，且對專題效果有幫助	布置空洞，且可能與專題無關
2. 故事主題	觀眾很容易明白	觀眾可以明白	觀眾有點明白	觀眾不明白
3. 動畫流暢性 (程式運作)	運作良好且具複雜性	運作良好	程式碼可能有錯且無法運作	無法運作
4. 多媒體（音樂、旁白等）	運用良好且效果佳	運用良好	有運用，但可能有錯或無法運作	未使用
5. 創意	非常有創意且獨特	有創意且獨特	有點創意且獨特	不具創意且不獨特

(二) Alice 程式設計學習態度問卷

本研究以「程式設計學習態度」問卷來瞭解兩班學生經歷過Alice學習後的態度。問卷題目包括6題李克特（Likert）類型的題目（內容請見表4，以1-5分代表非常不同意、不同意、普通、同意、非常同意），以及三題開放式題目，此三題為：「你在學習Alice動畫過程中，覺得最棒的三件事」、「你在學習Alice動畫過程中，覺得最糟糕的三件事」

及「你與別人合作一起用Alice說故事或製作動畫的經驗如何？你學到什麼？」，其中最後一題只有配對組的學生填答，個別學習組的學生並未填答，目的在於瞭解配對組學生在進行配對程式設計學習時是否遇到問題或困難。

肆、結果與討論

以下首先比較兩組學生的Alice動畫成果和學習Alice的態度，然後描述兩組學生對學習Alice的正、反兩方面的看法。

一、Alice 動畫成果比較

表3是兩組學生的Alice動畫作品評分統計結果。整體看來，配對學習組學生之平均分數高於個別學習組，可見Alice之學習成效，配對學習的效果顯著優於個別學習，與部分文獻（如Bishop-Clark et al., 2006; Howard et al., 2009）的結果一致。在Alice作品五個向度的分數中，除了「場景安排」分數，個別組比配對組略高以外，其他四個向度的分數，皆是配對組高於個別組，且在「故事主題」、「動畫流暢性」及「創意」方面已達到統計的顯著差異。觀察兩組學生在五個項目之表現，配對組最高分的項目是「創意」，可見配對學習共做的方式可以激發作品的創意；而個別組表現最佳的項目是「場景安排」；最差的項目則兩組一樣，皆是「多媒體」的運用，我們推測可能是學生怕麻煩所致，因為必須另外準備音樂檔、錄製旁白等，故運用的學生不多。

表3 Alice 動畫作品表現

評分項目	配對組		個別組		<i>t</i>	<i>p</i>
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>		
1. 場景安排	2.91	0.29	2.94	0.68	-0.24	0.813
2. 故事主題	3.00	0.74	2.52	0.79	2.49*	0.015
3. 動畫流暢性	3.00	0.43	2.58	0.64	3.28**	0.002
4. 多媒體	2.39	1.08	2.08	1.01	1.16	0.248
5. 創意	3.22	0.42	2.46	0.71	5.63***	0.000
平均	2.83	0.34	2.52	0.67	2.62*	0.011

* $p < .050$; ** $p < .010$; *** $p < .001$

二、Alice 學習態度比較

表4是兩組學生對Alice學習態度之統計。整體來說，除了第4題顯示個別組學生認為學習Alice有趣的比例顯著大於配對組外，其他五題，包括：學習Alice遇到挫折、學習Alice有信心、對Alice作品感到驕傲、往後還有興趣學習Alice以及有能力學習Alice等，兩組並無統計上的顯著差異。歸納而言，有以下四個現象：（一）配對組對於自己設計出來的Alice作品感覺驕傲（第1題）的比例大於個別組，與表2兩組學生Alice動畫作品表現之評分結果一致；（二）個別組學生遇到挫折（第2題）的比例略高於配對組，但未

達到顯著差異；(三)兩組學生對學習Alice動畫抱持信心(第3題)的比例皆不高，雖個別組(20%)略高於配對組(16%)，但整體來說，平均分數都只有3分；(四)個別組學生雖然在「學習Alice動畫有趣」(第4題)的分數顯著高於配對組，但在「往後還有興趣學習Alice」(第5題)的分數和「認為有能力學習程式設計」(第6題)兩項分數卻反而低於配對組。

表 4 Alice 學習態度

題目	組別	SA & A (%)	Neutral (%)	D & SD (%)	<i>M</i>	<i>t</i>
1. 我對自己(組)設計出來的 Alice 動畫感到驕傲	配對組	67	31	2	4.04	0.93
	個別組	66	26	8	3.86	
2. 我在學習 Alice 動畫時遇到挫折	配對組	68	28	4	3.78	-1.91
	個別組	78	20	2	4.04	
3. 我對學習 Alice 動畫抱持信心	配對組	16	70	14	3.00	0.00
	個別組	20	59	20	3.00	
4. 我覺得學習 Alice 動畫有趣	配對組	22	70	8	3.16	-2.25*
	個別組	42	50	6	3.47	
5. 往後我還有興趣再學習 Alice 其他的進階功能	配對組	26	54	20	2.98	1.37
	個別組	12	57	31	2.73	
6. 學習 Alice 後，我認為我有能力學習程式設計	配對組	40	40	20	3.22	1.19
	個別組	27	55	18	3.00	

註：SA(strongly agree)非常同意；A(agree)同意；Neutral 普通；D(disagree)不同意；SD(Strongly Disagree)非常不同意；* $p < .050$

在覺得學習Alice有趣方面，個別學習組高於配對學習組的原因，我們推測可能是個別學習組在製作動畫時，有比較多的機會操作Alice；而配對組一邊討論一邊製作時，只能有一人操作軟體，故個別學習組覺得Alice有趣的比例較配對組高。另外，在課堂觀察中我們發現，配對組學生比較會去挑戰高難度的動畫效果，但經過努力後，發現並不容易或皆可達成，故反而對學習Alice感到失望而覺得無趣。而兩組學生對往後再學習Alice的興趣皆不高(配對組vs.個別組：2.98 vs. 2.73)的原因可能與他們學習Alice遇到挫折有關。

三、學習 Alice 感到很棒的地方

從表3可知，不管配對組或個別組，大多數學生對自己製作出來的Alice作品感到驕傲。此外，在開放性問題「學習Alice最棒的三件事」中，學生也提到對於學習Alice，感到新鮮有趣、有成就感。以下舉出學生提到的描述：

- 「真的很感謝老師一步一腳印的教導我們如何編寫Alice的程式。看到自己從失敗中努力的成果，這已經不是用“感動”兩個字可以形容的了，是一種從心裡

油然而生的“成就感”」

- 「哈哈光是想主題就花了不少時間,後來做出來之後成就感真的很大,雖然重作的部分很多,也要一直注意有沒有做錯的地方,細節的部分也花費了很多心思。」
- 「總而言之,這次Alice也讓我學到不少東西,可以自己做動畫真的很有趣呢!」
- 「做Alice,其實還滿有趣的,雖然過程都是怨恨聲,為什麼要做這個?」
- 「這次的作業,雖然很費時,構思也要很多,但是我覺得一起討論的感覺其實用心體會真的蠻好的。一起對這件作業提出各自的想法,再將2個結合起來成為新的元素。看著作品,成就感在心裡慢慢滋生,讚!!」
- 「很开心和我的同學一組並一起完成Alice!!~」
- 「雖然在製作的過程中出了點小小意外,但是我們還是很順利的把Alice給做完了,這真的是一個很特別的經驗!」
- 「製作過程中常常被那個很搞怪的動作弄到笑翻!☺這可能是我在做Alice中的唯一樂趣吧!」

四、學習 Alice 時遇到的問題

在開放式問題「學習Alice最糟糕的三件事」中,學生提到學習Alice時遇到的困難為:

(一)英文不熟悉

部分學生反應Alice英文的介面影響了他們的學習。他們在問卷填答中提到:「我覺得這是一個很有趣的程式,可以做出自己喜歡的動畫,但是唯一有缺點的地方就是參雜著太多英文的作法,這使英文不太好的我非常困擾。還好有神人**為我們找出解決方法,最後才能做出這完美的動畫!!!!YA!!!」。

(二)沒有信心

經調查,幾乎所有的學生都是第一次接觸程式設計課程,因此對程式學習缺乏信心。學生提到:「我們都是第一次碰觸有關於類型的程式,一開始聽到要做有關Alice我們都兩顆頭四個大」、「我們明明就是護理科,幹嘛要寫程式!」可見學生對程式設計學習的信心明顯不足。對於Alice學習沒有信心的現象,不管是配對組或個別學習組皆有學生提到,故表3第3題顯示兩組學生的信心平均分數只有3分。

(三)電腦當機

學生提到:「電腦很爛常常當機,沒有存檔,就要一切重頭開始,但當屬於自己的程式寫出來,會有滿滿的成就感,與夥伴們也增進不少感情。」此電腦當機和存檔的問題,我們發現一部份是輸入中文所造成,加上程式不穩定之故,後續的Alice版本已改善許多,且已支援上傳到Youtube。

(四)不易做出精緻的動畫

由於課程時間只有五週,學生在短時間內很難學到高階的程式技巧,加上Alice軟體雖標榜可做出3D動畫,但主要是讓學生學習物件導向程式設計的觀念,因此一些精緻化的動作,Alice並不支援;但學生可能還缺乏此正確觀念(雖然課堂上已告知)或想要自

我挑戰的決心，所以一直想要做出她們理想中的動畫效果，但也因此而落空。學生提到：「我覺得這個程式還是有點死板，那個坐下的動作一直做不出來，到最後我們就放棄了！」因此，未來如能加長課程實施時間，相信學生程式設計的功力可提升，也能製作出更逼真、更棒的動畫了。

伍、結論與建議

根據前一節之研究結果及討論，本研究歸納出以下結論和未來教學與研究建議。

一、配對學習可促進 Alice 學習的表現

本研究發現配對組的Alice作品，在「故事主題」、「動畫流暢性」及「創意」等三個向度皆顯著優於個別組的學生，可見配對方式可以促進Alice學習的成效，與Bishop-Clark等（2006）及Howard等（2009）之研究中，發現配對方式對提高學習成效有幫助的結果一致，但這些研究顯示配對組學生學習興趣更高之結果，則與本研究不一致。另外，本研究並未如Kelleher（2006）之研究中，發現配對組兩人在協商故事情節時遇到困難，因而導致對話中止之情形。此現象可從問卷開放性問題之第三題：「你與別人合作一起用Alice說故事或製作動畫的經驗如何？你學到什麼？」，95%以上配對組的學生之回答皆是正向的話語（如：很棒、開心、合作愉快、學到很多等）而得到驗證。

二、學生對學習 Alice 具成就感，但也面臨一些問題

不管配對組或個別學習組，大部分學生對Alice的學習皆感到新鮮有趣，對自己可以學習做動畫也覺得有成就感。但是，兩組學生在學習過程中皆面臨了一些問題，包括：對英文介面和語法不熟悉、電腦有時當機、以及不易做出精緻動畫，我們推測是由於學生對程式設計學習信心不足、以及對Alice軟體的定位功能不清楚所致。

三、未來教學與研究建議

本研究發現配對組學生的Alice作品成果雖然高於個別組，但個別組覺得Alice有趣的比例卻高於配對組。因此，本研究建議未來的課程規劃，可以先讓學生個別操作Alice後，再讓學生兩人一組進行配對學習，也可根據學生的學習風格進行安排，以順應學生的喜好，達到最佳的學習效果。另外，由於課程時間只有五週，本研究只針對學生一次的Alice作品進行學習成效評估，且主要針對其動畫效果進行評分，未從程式設計概念，如物件、變數、陣列、迴圈、函數、方法、同時執行、照順序執行等內容進行學習成效的檢驗，未來可加入此部分的評分，以引導學生進入更高階的程式設計學習。

參考文獻

- 尹玫君 (1991)。電腦程式設計能力與認知能力相關之研究。《台南師院學報》，24，39-54。
- 王鼎中、丘聖光、林淑玲、梅文慧、林美娟 (2009)。創新程式設計課程與教學模式之研發。《科學教育月刊》，321，2-11。
- 黃建庭 (2003)。《配對程式設計的學習成效評估—應用演算法模擬軟體於迴圈的學習》。國立台灣師範大學資訊教育研究所碩士論文，未出版。
- Anabela, G., & Mendes, A. J. (2007). Learning to program - difficulties and solutions. *Proceedings of 10th International Conference on Engineering Education*, 283-287.
- Beck, K. (1999). *Extreme programming explained: Embrace change*. Boston: Addison-Wesley Longman Publishing Co., Inc.
- Bishop-Clark, C., Courte, J., & Howard, E. V. (2006). Programming in pairs with Alice to improve confidence, enjoyment, and achievement. *Journal of Educational Computing Research*, 34(2), 213-228.
- Brought, G., Edy, M., & Wahls, T. (2008). The effects of pair-programming on individual programming skill. *ACM SIGCSE Bulletin - SIGCSE 08*, 40(1), 200-204.
- Cao, L., & Xu, P. (2005). Activity patterns of pair programming. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 3, 1-10.
- Chong, J., Plummer, R., Leifer, L., Klemmer, S. R., Eris, O., & Toye, P. (2005). Pair programming: when and why it works. *The 17th Workshop of the Psychology of Programming Interest Group PPIG17*, 43-48.
- Copper, S., Dann, W., & Pausch, R. (2000a, November). Developing algorithmic thinking with Alice. *Information Systems Educators Conference*, Philadelphia, PA.
- Copper, S., Dann, W., & Pausch, R. (2000b). Alice: a 3-D tool for introductory programming concepts. *Journal of Computing Sciences in Colleges*, 15 (5), 107-116.
- Flor, V. N. (1998). Side-by-side collaboration: a case study. *International Journal of Human-Computer Studies*, 49(3), 201-222.
- Howard, E. V., Evans, D., Courte, J., & Bishop-Clark, C. (2009). A qualitative look at Alice and pair-Programming. *Information Systems Education Journal*, 7 (80), 1-10.
- Jenkins, T. (2002). On the difficulty of learning to program. *Proceedings of 3rd LTSN-ICS Conference*, 53-58.
- Kelleher, C. (2006). *Motivating programming: using storytelling to make computer programming attractive to middle school girls*. Unpublished doctoral dissertation, Carnegie Mellon University.
- Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: a survey of programming environments and languages for novice programmers. *ACM Computing Surveys*, 37(2), 83-137.
- Linn, M. C. (1985). The cognitive consequences of programming instruction in classrooms.

Educational Researcher, 14(5), 14-29.

- Mayer, R. E., Dyck, J. L., & Vilberg, W. (1986). Learning to program and learning to think: what's the connection? *Communications of the ACM*, 29(7), 605-610.
- McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2002). The effects of pair-programming on performance in an introductory programming course. *Proceedings of the 33rd SIGCSE technical symposium on Computer science education*, 38-42.
- Preston, D. (2005). Pair programming as a model of collaborative learning: a review of the research. *Journal of Computing Sciences in Colleges*, 20(4), 39-45.
- Simon, B., & Hanks, B. (2007). First-year students' impressions of pair programming in CS1. *ACM Journal on Educational Resources in Computing*, 7(4), 5:1-5:28.
- Williams, L., & Kessler, R. (2002). *Pair programming illuminated*. Boston: Addison-Wesley.
- Williams, L., & Upchurch, R. (2001). In support of student pair programming. *SIGCSE Conference on Computer Science Education*, 327-331.
- Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In support of pair programming in the introductory computer science course. *Computer Science Education*, 12(3), 197-212.