

專題式程式設計教學對國小學童問題 解決歷程之研究

張玟慧

博士候選人
國立臺北教育大學
E-mail: 0805@gmail.com

吳佳娣

教師
新北市莒光國小
E-mail：charity1113s@gmail.com

陳彤宣

博士候選人
國立臺北教育大學

摘要

本研究目的在探討專題式程式設計課程，對學生問題解決能力與解題態度之影響。研究方法為準實驗研究法，進程式設計教學課程。實驗組以專題導向學習策略進行教學，對照組則採編序式教學，進行為期 16 週之程式設計教學。研究參與者為新北市某國小四年級學生，共 323 人。本研究分為量化資料與質性資料。量化資料以卡方考驗檢定與 T 檢定，分析實驗組與對照組間的差異；質性資料來自教學紀錄與實驗組焦點訪談紀錄。研究結果顯示：專題式程式設計教學對國小學童的學習成就、解題態度、對程式設計的看法與問題解決能力有正面的影響，使用專題導向學習的學生，會展現出更高的執行計畫以及反省回顧能力。教師在專題進行中，能隨時監督學生狀況，並給予支持、引導其針對眼前問題作分析，搜尋解決的方法，有助於學生將在專題導向課程中，習得經驗與問題解決技巧，運用在往後的學習與日常生活之中。

關鍵字：專題導向學習、Scratch 程式設計軟體、程式設計、問題解決、解

題態度



CACET
中華資訊與科技教育學會

壹、緒論

許多研究指出，透過學習程式語言可以幫助學生培養問題解決的能力（Brown, Fontecchio, Fromm, Garbarine, Kusic, & Mongan, 2008；Mayer & Wittrock, 2006；Dyck, Mayer, & Vilberg, 1986）。Barker、Lao、Reynolds與Wu（2008）認為兒童可以透過學習程式設計，增進邏輯思考、組織、判斷與問題解決的能力；Shafto（1986）發現學習程式設計有助於孩子們學習科學與數學，因為程式設計可讓孩子們學習如何以清楚、正確、又可執行與驗證的形式來描述解題方法。

為了達到解決問題的目標，在程式設計教學設計上應該「以問題解決為導向，再導入需要使用的相關指令與敘述」（吳正己、何榮桂，1998；吳正己、林凱胤，1997）。許多研究也指出（Chao, 2016；Lin & Pedersen, 2003；Salomon & Perkins, 1987；Seidman, 1988；Shaw, 1986），程式設計與問題解決的關係密不可分，必須以明確的問題解決方法來教授程式設計，方能利於學習者將所習得的問題解決技能，類化到其它相關的知識領域。以解決問題為目標的教學模式可分為問題導向與專題導向，其皆強調在真實世界的環境中學習，以學生成效為依歸。兩者相異之處在於：專題導向學習是較為行動傾向的，學生需主動設計與自己興趣相符的作品、並持續增進自己的表現；問題導向學習則要求學生專注在一個待解決的問題上。Bouhuijs、Perrenet與Smits（2000）則認為專題導向學習比問題導向學習更需要知識的「應用」。由於資訊與網路的普及，學生幾乎都有玩電腦遊戲的經驗，因此在選擇專題內容上，以貼近學生興趣與生活經驗的遊戲製作為主題。Hayes（2008）認為透過製作遊戲，可精熟學習者的資訊科技能力（例如：使用繪圖工具、影像處理軟體、程式設計等等），讓其在資訊發達的社會網絡中，不再只是資訊的消費者，而能成為有生產力、訓練有素的使用者。

本研究基於提升學生問題解決能力，加強知識應用的研究動機，以設計遊戲的專題導向學習，進程式設計教學之研究。本研究目的在了解以專題導向學習（Project-based Learning）進程式設計教學，對學生學習程式、設計成效、程式設計態度，以及提升問題解決各階段能力的影響。

貳、文獻探討

一、專題導向學習

專題導向學習（Project-based learning）是一個以專案形式組織學習的模型，為建構取向的學習方法。其融合了建構主義、合作學習與情境學習之基礎，強調教師為學習輔助角色，安排學習任務（專題計畫）要求學生產出成品，使學生能夠從中主動學習、主動參與問題解決，以在學習任務中建構自己知識的一種教學

方法（邱貴發，1996；Thomas, 2000）。專題導向學習可以提升學生的自我調節能力（Lin & Tsai, 2016），並進一步提升問題解決能力（Tanweer, Suresh, & Sundararajan, 2016）。學習者決定如何解決問題與需要的活動，蒐集多元資料整合、分析和擷取知識，並產出作品。專題導向學習的內容可以是單一學科或跨學科領域，且專案的製作可以獨立或合作方式完成，時間可能從幾週到一整年（Mills & Treagust, 2003）。儘管許多人將「問題導向學習」與「專題導向學習」二詞經常交替使用，顯示兩者之間的灰色地帶，但 Hadgraft, Ilic 與 Scott（2003）認為專題導向較適合使用於數學、物理或工程學等領域。由於這些學科是具有階層知識結構的，如果缺少一個基本的部分，會導致學習其他概念的失敗，必須依照一定的學習順序，因此在知識與技能的提供上，需要教學者較多的主導。綜合上述，可知專題導向學習的特色有：（1）以學習者為中心；（2）設計學生在真實世界的情境中學習；（3）強調實作評量；（4）學生需主動設計與自己興趣相符的作品、參與問題解決，並持續增進自己的表現；（5）適合使用於具階層知識結構的學科領域。

由於資訊與網路的普及，學生幾乎都有玩電腦遊戲的經驗，因此在選擇專題內容上，以貼近學生興趣與生活經驗的遊戲製作為主題。Hayes（2008）認為透過製作遊戲，可精熟學習者的資訊科技能力（例如：使用繪圖工具、影像處理軟體、程式設計等等），讓其在資訊發達的社會網絡中，不再只是資訊的消費者，而能成為有生產力、訓練有素的使用者。另外，遊戲本身就是由許多小問題，隨意連結而成的大問題（Kiili, 2007），因此在遊戲的過程中，亦能經歷「遭遇問題」、「思考決策」、「執行計畫」與「回顧」的問題解決歷程。應用以遊戲為主題的專題導向學習於程式設計，不只能透過實作的方式評量學習，學生亦能根據自己的經驗來設計有興趣的主題，並在測試遊戲的過程中，反覆地進行問題解決，除精熟自己的資訊科技能力外，也培養問題解決的能力。

綜合上述，在課程設計上以專題導向學習為基礎，透過解決與生活經驗相關以及符合自己興趣的問題學習程式設計，探討其對學生學習程式設計的成效、態度與問題解決能力的影響。

二、程式設計教學策略

程式設計與許多科技領域有關，但教導年輕學生學習程式設計一直是一項困難的挑戰，而在基礎課程中就遭遇困難的結果，使得程式設計課程的退選率一直居高不下。程式設計能帶來一些重要的好處，例如：擴增使用電腦創作與表達自我的能力，也擴大學習者能學習的範圍。程式設計支援「計算思考」的發展，幫助學習重要的問題解決和設計策略，並能運用在其他領域（Wing, 2006）。Jeschke、Kato 和 Knipping 在 2007 年於國小進行 LEGO NXT Mindstorms 課程，其

中包括分析階段、設計階段、技術執行階段、測試階段和最終展示階段。顯示這樣的活動能支持學生建立基礎科技知識，培養問題解決方法。然而個人電腦從七〇年代晚期推出至今，已經成為許多人生活中的一部份，但電腦程式設計依舊被視為狹窄的專業技術活動，只適合少數人。原因在於傳統的程式設計軟體使用困難、程式設計的教學活動使用大量的抽象概念與數學問題，與學習者的生活經驗無關，而不透明的程式執行過程、文字型態的輸出，也容易使學習者感到困惑與無趣。舉辦與電腦科學相關比賽的人員也發現，參與競賽的年輕學生動機低落，尤其在參賽年齡日益下降時更加明顯 (Idlbi, 2009)。

Copper、Dann 與 Pausch (2003) 認為程式設計的四大困難分別為語法、無法看到輸出、缺乏學習動機和難以理解的邏輯與設計技巧。

綜合上述，可整理出程式設計的困難在於：抽象的程式概念、難以記憶的語法、不透明且無趣的執行結果，與學習動機的缺乏。因此，本研究在程式設計教學策略上，選擇視覺化的程式設計工具，配合貼近學生生活經驗的範例與題目，減少學習程式設計之困難。

三、問題解決模式與程式設計解題歷程

當面臨一個問題且找不到任何明顯的解決方法，我們會進入一個認知處理過程。這個為了找到解答而產生的認知處理過程，就是「問題解決」(Mayer & Witrock, 2006)。雖然問題解決是一個內在認知的歷程，但已有許多學者提出具體的論點，將問題解決的歷程明確地依次列出，包括杜威的問題解決五步驟：遭遇問題、界定問題、發展假設、驗證假設、應用；Polya 發展的問題解決模式：了解問題、想出計畫、執行計畫、回顧。程式設計是運用程式語言撰寫程式以解決問題。程式設計包括四個主要的步驟：了解問題的需求、擬定解題的計畫、撰寫程式碼，以及測試與除錯 (吳正己、林凱胤, 1997)，與問題解決模式類似。學習者能依循這樣的模式，有系統地思考、規劃、執行與檢視結果來解決問題，方能增進自己的概念認知。Kiili (2007) 認為遊戲即是由許多小問題，隨意組合而成的大問題，並研究玩家在遊戲中所經歷的學習歷程，發現玩家在遊戲時所經歷的「形成遊戲策略、實驗遊戲策略、反思」過程，能幫助玩家將理論性的知識實際應用，並達到更高的理解與擴增領域知識。提出類似論點的還有 Ahlers, Driskell 與 Garris (2002) 的判斷、行為、回饋。各解題模式與歷程的比較表 2-1：

表 2-1 各解題模式與歷程比較表

Polya 解題模式	程式設計解題歷程	Ahlers, Driskell and Garris
了解問題	了解問題	判斷
想出計畫	擬定解題的計畫	行為
執行計畫	撰寫程式碼	行為



* 研究者整理

綜合比較多個關於解決問題的歷程與模型，發現與程式設計的解題歷程類似，因此本研究期望透過學習程式設計，使學生經歷程式設計解題歷程，提升其解決問題的能力。

參、研究方法與設計

一、研究對象

本研究參與者為新北市某國小之 10 個班級的四年級學生，共 323 人。研究參與者根據學期總成績，經 T 分數調整後進行全校排名，以 S 型常態編班方式分班，因此各班級學生起點能力差異不大。研究參與學生從三年級開始接受資訊課程，具備 Windows 作業系統與網際網路之基本知識與操作技能，學習過文書處理軟體 word，能使用文書處理軟體製作表格、插入圖片、輸入文字內容和變更其字型與色彩，亦具備將作品儲存在指定資料夾之能力，但皆無程式設計之經驗，為程式設計初學者，學生起始行為相當。

研究參與者隨機以班級為單位分為實驗組（163 人）與對照組（160 人）。在為期 16 週的程式設計課程中，實驗組進行專題導向學習，對照組接受編序教學。

二、實驗設計

本研究旨在探討專題式程式設計教學對學生問題解決能力與解題態度之影響，自變項為教學設計，依變項為學生的問題解決能力與解題態度。為增加研究之嚴謹度，本研究之實驗組與對照組為相同教學者進行教學，教學時間皆為 16 週，與學生皆為同一所學校之學生，其學區之學生社經背景具一定程度的相似度。

（一）教學設計

1. 分析

（1）教學目標

為使程式設計初學者能在輕鬆、有趣的環境下學習程式設計，建立思考邏輯，故課程內容以製作互動式遊戲為主，並使用圖像與聲音等多媒體檔案，提升學習意義與學習者的學習動機。

（2）學習者

學習者為國小四年級之學生，具備電腦基本操作能力與網際網路概念，無程式設計之經驗，為程式設計初學者。學習者屬於具體運思期，故在課程設計上，以貼近其生活經驗為原則，設計學生容易理

解的題目讓學生解題，並展示範例執行的結果，使學習者能清楚了解題目。

2. 設計：

課程共分為三個階段，共六個單元，從設計並製作遊戲的過程中，學習抽象的程式概念。在每一單元開始前，先說明學習重點，然後帶領學生了解題目，試著將其分解為許多小步驟，再從程式區塊中選擇適當的程式碼，依正確的順序組合在一起，並測試完成後的執行結果。兩組都先進行教學活動，介紹程式設計概念與結構。對照組在每一階段的課程結束後，提供「自己動手作」的練習題，鼓勵學生針對問題，將所學的知識重新組織；實驗組則製作自行設計之遊戲，並透過測試遊戲進行程式的修正，教師從旁提供引導與協助。

3. 開發：

由於學習者為程式設計初學者，課程從介紹最基礎的程式概念開始，題目要以接近學習者的生活經驗為原則，必要時配合插圖說明，敘述以簡潔但詳細的方式描述。在每一階段課程開始前，說明學習重點，並回顧過去經驗或已經學習的程式概念；在示範操作時，縮減步驟使學習者能持續跟上進度，並藉由反覆的類似操作，加深學習者印象。

4. 執行：

(1) 時間

本研究設計之課程在九十八學年度第二學期實施，研究對象為研究者任教學校之四年級學生，實驗教學時間共 16 週，每週一堂課，共 640 分鐘。

(2) 環境與設備

實驗進行的地點在研究者任教學校之電腦教室，教學進行時，教師電腦之畫面可透過教學廣播系統廣播至學生電腦螢幕上，使學生能清楚看到操作過程；學生操作時為一人一機，有充分練習之機會。

(3) 授課方式與教材呈現

授課方式以講述法為主，教師先進行操作示範與講解，再讓學生自行練習。教材呈現的方式有圖解講義、學習單和操作畫面錄影檔，以超連結方式放置在網頁上，學生可直接點選下載，或在上課時利用 HFS (http file server 虛擬檔案伺服器) 從教師電腦下載檔案。圖解講義可以輔助學生在自行操作時，視需要隨時閱讀，降低工作記憶的負荷；學習單上列有實作練習題說明或紙筆測驗，學生可利用 5 到 10 分鐘進行自我檢測，教師也能藉此發現學生對課程內容的了解是否正確；透過操作畫面錄影檔，學生可在家或利用課餘時間觀看複習，掌握自我的學習步調。

5. 評鑑：



實驗教學結束後，進程式設計成就測驗、程式設計態度問卷與問題解決能力測驗，以分別了解不同教學策略對學生程式設計的學習成效、學習興趣，與問題解決各階段能力的影響為何。

(二) 課程內容

對照組與實驗組的課程內容，分別簡介如下：

1. 對照組：

對照組進行編序教學，教師先進行操作示範與講解概念，讓學生自行練習後，提供問題範例讓學生進行解題。教學時間為 16 堂課，每堂課 40 分鐘，共規劃四個單元，兩次形成性評量，如表 3-1：

表 3-1 對照組教學時程與內容規劃

節數	單元名稱	教學目標	教學內容
2	Scratch 初體驗	熟悉 Scratch 操作介面	在 Scratch 中練習組合指令積木、執行程式碼與儲存檔案
2	角色動一動	學習程式的循序概念	介紹「控制」、「動作」、「外觀」、「聲音」指令，以及使用 2~3 個指令積木組合第一個程式
3	角色移動控制	學習判斷結構	介紹迴圈、判斷與條件的程式概念
3	評量活動一： 電流迷宮	了解並能應用程式的判斷結構	提供問題讓學生以程式設計進行解題，重新組織所學知識
3	倒數計時器	學習程式的迴圈與變數概念	介紹 for 迴圈、「變數」指令和組合程式碼
3	評量活動二： 水果雨	了解並能應用程式的迴圈、變數概念	提供問題讓學生以程式設計進行解題，重新組織所學知識

2. 實驗組：

實驗組進行專題導向學習，教師先引導學生分析構成遊戲的基本元素，並自己設計一個遊戲計畫。教學活動根據 PBL 步驟，分為四階段：開始階段、專題計畫階段、專題執行階段和完成階段。開始階段包括定義時間表、評量方式、確認教學資源與先備知識；專題計畫階段包括學生初步確認專題細節、規劃自己的計畫表、教師提供回饋與教學；專題執行階段包括學生根據自己的進度製作專題、教師在旁監督與提供協助；完成階段包括遊戲分享、同儕回饋與繳交最後作品。教學時間為 16 堂課，每堂課 40 分鐘，共規劃六個單元，如表 3-2：

表 3-2 實驗組教學時程與內容規劃

節數	單元名稱	教學目標	教學內容	PBL 階段
2	我的遊戲計畫	設計自己的遊戲專題	分析遊戲中的組成元素，並讓學生設計一個遊戲（教材如圖 3-5）	開始階段： 專題計畫階段



2	Scratch 初體驗	熟悉 Scratch 操作介面	在 Scratch 中練習組合指令積木、執行程式碼與儲存檔案	
2	循序概念	學習程式的循序概念	介紹「控制」、「動作」、「外觀」、「聲音」指令，以及使用 2~3 個指令積木組合第一個程式	
2	判斷與條件	學習判斷結構	介紹判斷與條件的程式概念	專題執行階段
3	遊戲製作 (一)	學生主動學習、設計遊戲、分享與測試	學生主動學習、設計遊戲，應用習得知識並透過遊戲測試除錯	
3	迴圈與變數	學習迴圈與變數概念	介紹迴圈和「變數」指令	
2	遊戲製作 (二)	學生主動學習、設計遊戲、分享與測試	學生主動學習、設計遊戲，應用習得知識並透過遊戲測試進行除錯	
1	作品觀摩	學生互相觀摩、試玩作品，提供同儕回饋	每一學生需邀請至少兩位同學試玩自己設計的遊戲，並請對方給予建議與讚美	完成階段

三、研究工具

本研究蒐集的資料為量化資料與質性資料。量化資料為程式設計成就測驗成績、程式設計態度問卷與問題解決測驗之填答結果，使用卡方檢定與 T 檢定法，分析對照組與實驗組在程式設計學習成效、程式設計學習態度，以及在問題解決歷程中所提升的能力是否達到顯著差異。質性資料為教學紀錄與焦點訪談逐字稿，用於量化資料所得結果可能原因之佐證。

研究環境為研究者任教學校的電腦教室，內有 36 台學生用電腦以及一台教師用電腦，每台電腦具基本配備：液晶螢幕、滑鼠、鍵盤，安裝 Windows XP 作業系統，並可透過新北市網路連結網際網路。教師可利用廣播系統將電腦畫面廣播至學生螢幕，進行教學示範與解說；學生平時上課為一人一機，可充分練習與操作。

本研究採準實驗研究法，研究工具有 Scratch 程式設計軟體、對照組與實驗組課程簡案、上課講義、遊戲設計學習單、程式設計成就測驗、程式設計態度問卷、問題解決測驗前後測、教學紀錄表與訪談紀錄表，分述如下：

(一) Scratch 程式設計軟體

Scratch 是美國麻省理工學院 (MIT) 媒體實驗室開發的一套視覺化圖形介面的程式設計軟體，採用拖曳的方式 (drag-and-drop) 組合指令積木，支援豐富的多媒體素材，讓使用者能自由而輕鬆地創作各式各樣的專案，如：動畫故事、互動遊戲、音樂作品等。Scratch 的操作簡單，適合容易對手冊說明過多感到不耐的青少年使用，且立即的視覺回饋不僅能具體化抽象的程式概念，也能提高學習者的興趣。Scratch 程式設計軟體畫面如圖 3-1：

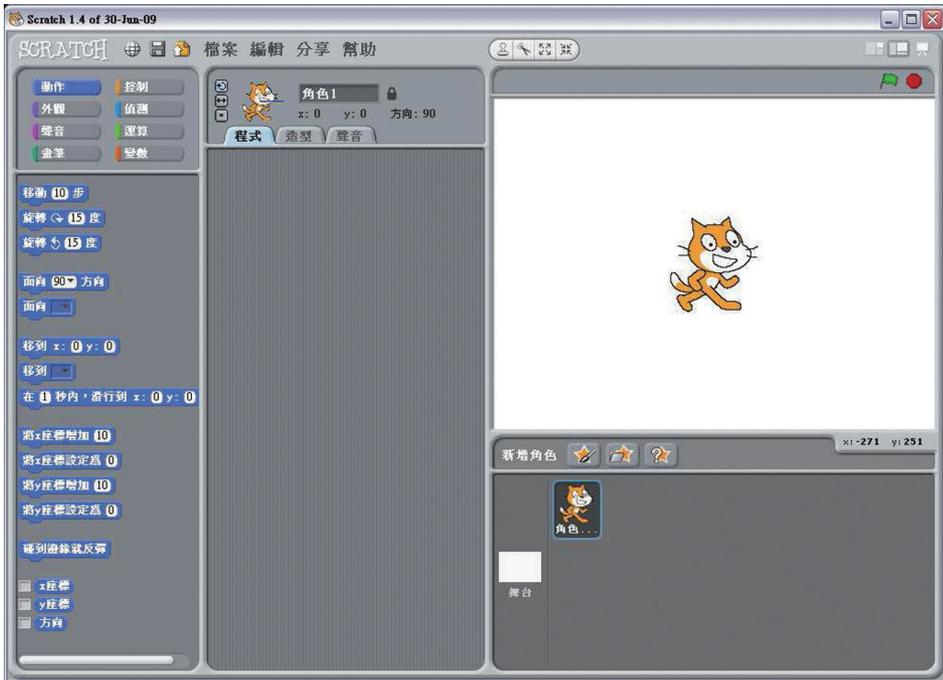


圖 3-1 Scratch 程式設計軟體畫面

Scratch 程式設計軟體以拖曳的方式組合指令積木為程式區塊，指令積木依照不同屬性分為不同的顏色和形狀，且只有在有意義的情況下才能組合在一起，可減少初學者常見的語法錯誤。Scratch 指令積木與程式區塊如圖 3-2：

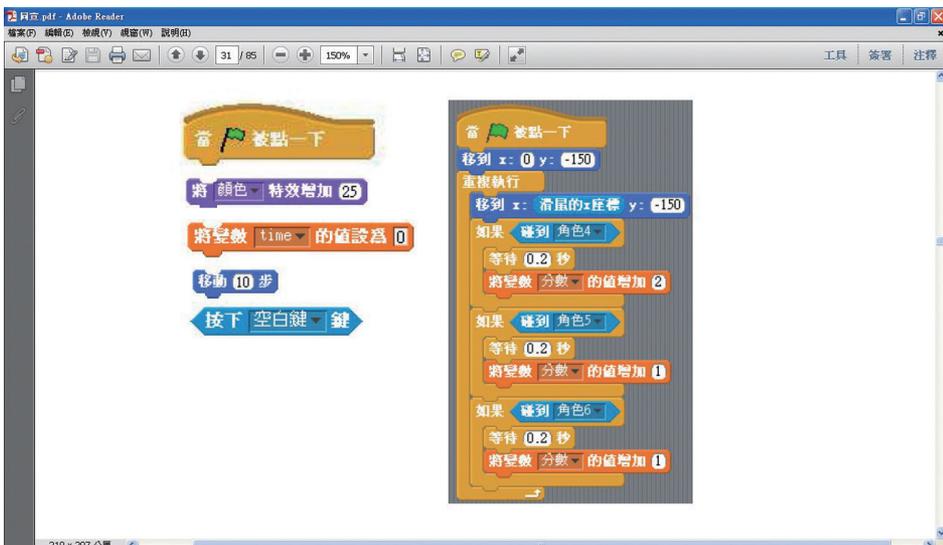


圖 3-2 Scratch 指令積木與程式區塊

(二) 課程簡案



分為對照組與實驗組的教學簡案，說明各單元與活動進行的時間、教學目標、教學步驟與使用之教材。以下節錄對照組單元一之教學簡案，如圖 3-3：

單元名稱	Scratch 初體驗
教學時間	2 節，共 80 分鐘
教學目標	1. 認識 Scratch 操作介面 2. 能拖曳各指令積木，放置到程式組合區 3. 能控制遊戲開始與停止 4. 能將檔案儲存到指定位置
教學活動步驟	1. 如何打開 Scratch 2. 介紹 Scratch 操作介面 3. 說明指令積木的組合方式與條件 4. 練習儲存檔案到自己的資料夾
使用教材	1. Scratch 程式設計軟體 2. Scratch 操作介面講義

圖 3-3 對照組單元一「Scratch 初體驗」教學簡案範例

(三) 上課講義

程式設計教學課程配合自編講義，讓學生上課時除了聆聽教師講解與示範外，能搭配講義內容以減輕學生記憶的負荷，並在自行操作練習時有所輔助。以下節錄 Scratch 程式設計軟體操作介面講義（如圖 3-4）與遊戲組成分說明講義（如圖 3-5）：



圖 3-4 Scratch 程式設計軟體操作介面講義

遊戲組成說明

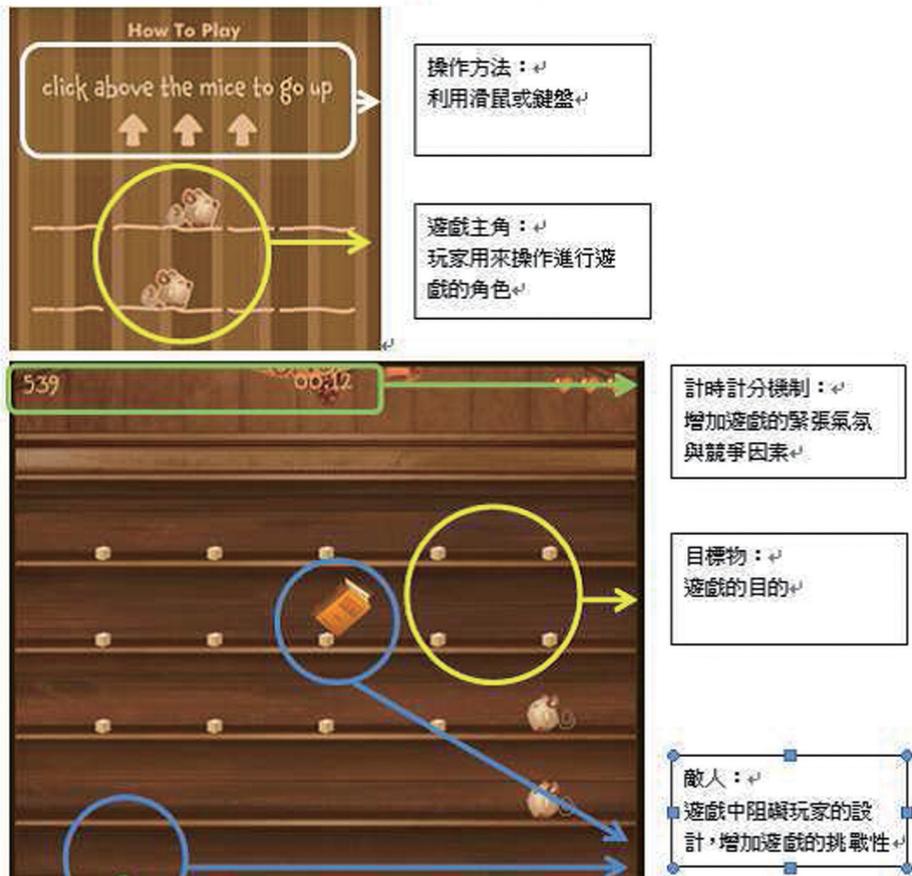


圖 3-5 遊戲組成成分說明講義

(四) 遊戲設計學習單

遊戲設計學習單配合實驗組課程的「我的遊戲計畫」單元進行，協助學生在設計遊戲時，能包含遊戲的主角、操作方法與目標等，並詳細寫出遊戲的規則，以作為未來製作遊戲時的依據與藍圖。

(五) 程式設計成就測驗

本研究的成就測驗以筆試與實際操作的方式進行，內容包含循序、判斷及重複等重要觀念，题目的敘述方式則與講義相似，皆是以 Scratch 程式設計軟體所構成的環境為主。全部題目共有六題，總分為 32 分，其目的在於瞭解學生的學習成效，並作為實驗比較之用。試卷依測驗內容「理解程式」與「設計程式」兩部分，其內容簡述如下：

1. 理解程式

理解程式題共有三題，總分為 6 分。學生根據題目的程式碼，選擇正確的輸出或寫出執行情況，藉此測驗學生對程式的理解程度。理解程式部份的第一題測驗學生對於循序概念的理解，全對得 3 分，第二題測驗學生對於重複結構與判斷的概念，全對得 2 分，第三題測驗學生對於變數與重複結構的概念，全對得 1 分。

部分題目如圖 3-6：

一、請根據下列程式碼，排出正確的執行順序。



程式執行的順序是：_____、_____、_____。

(甲) 位置在 X 座標 100，y 座標 50 (乙) 主角的大小變大

(丙) 主角的樣子變為

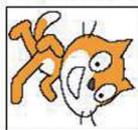


圖 3-6 程式設計成就測驗理解部分題目

2. 設計程式

程式設計題共分為三題，總分為 26 分。學生根據題目的描述，組合出程式並產出相同的輸出，藉此測驗學生的程式設計能力。設計程式部分的第一題為重複結構與判斷結構的應用，學生依據題目的描述組合程式，使角色能有相同的執行結果。選擇一塊正確的積木得 1 分，組合正確得 2 分，總分為 8 分；第二題為循序概念的應用，學生依據題目的描述組合程式，使角色能有相同的執行結果。選擇一塊正確的積木得 1 分，組合正確得 2 分，總分為 9 分；第三題為變數與重複結構的應用，學生依據題目的描述組合程式，使角色能有相同的執行結果。選擇一塊正確的積木得 1 分，組合正確得 2 分，總分為 9 分。

部分題目如圖 3-7：

四、請根據題目敘述，組合出程式碼：
遊戲執行時，如果按下滑鼠左鍵，將角色移到 x 座標為 0、y 座標為 0 的地方。

圖 3-7 程式設計部分題目

(六) Scratch 程式設計態度問卷

程式設計態度問卷內容為了解學生對於程式設計的看法，題目共有 13 題，以李克特式五點量表方式填答，其內容包括：對程式設計的看法、解題態度、學習動機、對程式設計的信心與學習興趣等。問卷整體信度係數 Alpha 值為 0.868，

達顯著水準，邀請三位國小資訊教師進行檢驗，建立專家效度。

(七) 問題解決測驗

「問題解決測驗」為潘怡吟(2002)參考林麗惠於2000年所編製的「問題解決測驗」，分為前測與後測，包含了圖片16幅與20個問題。評分者間信度係數為0.97，內部一致性信度為0.81，均達顯著水準依據。「問題解決測驗」以團體施測的紙筆測驗形式實施。為了解學生在經過專題導向學習程式設計教學，對其問題解決各歷程能力的影響為何，測驗內容分為四部分，分別為「了解問題」、「擬訂計畫」、「執行計畫」與「反省回顧」方面的能力。

(八) 訪談紀錄表

在教學實驗結束後一週內，抽選實驗組每班男、女學生各一名，共10名學生進行訪談。訪談目的在了解實驗組學生於進行遊戲設計專題時，解決問題的作法與思考過程，訪談的內容將會以錄音方式記錄下來，並由研究者轉換為逐字稿。

「資料編碼」以三碼來表示，第一碼代表資料來源，第二碼代表該資料中的頁碼，第三碼代表該資料所在行數。例如：編碼“(訪:1:2)”代表資料來源為訪問，位於資料頁中的第1頁第2行。

(九) 教學紀錄表

在進行教學實驗的16週間，研究者會在該天課程結束後，依照課程內容與課堂上觀察到的情形，撰寫教學紀錄，除作為教學省思之外，也作為研究結果之參考與推論依據。

教學紀錄之編碼以四碼來表示，第一碼代表資料來源，第二碼代表該資料中的頁碼，第三碼代表該資料所在行數，第四碼代表該紀錄的週次。例如：編碼“(教:12:15:w11)”代表資料來源為教學紀錄第12頁的第15行，為第11週之教學紀錄。

肆、研究結果

本研究目的在了解以專題導向學習進行程式設計教學，是否能有效提升學生在程式設計學習成就、態度與問題解決能力上的表現。研究結果分析如下：

一、程式設計成就

程式設計成就測驗共有六題，分為理解與實作兩部分，總分為32分。理解部分有三題，測驗學生循序、重複與判斷、以及迴圈與變數的概念，總分為6分；實作部分有三題，測驗學生重複與判斷、循序、以及迴圈與變數的概念，選擇正

確的指令積木即得 1 分，組合正確得 2 分，總分為 26 分。

測驗結果以 T 檢定作分析，如表 4-1。實驗組平均得分為 26.1、標準差 4.4，對照組平均得分為 23.2、標準差 5.5，p 值為 0.000，兩者達顯著差異 ($p < 0.01$)，實驗組成績優於對照組。程式理解部分，實驗組平均得分為 5.1、標準差 1.2，對照組平均得分為 5.2、標準差 1.2，兩者無顯著差異；程式實作部分，實驗組平均得分為 21.0、標準差 3.8，對照組平均得分為 18.0、標準差 4.9，p 值為 0.000，兩者達顯著差異 ($p < 0.01$)，實驗組成績優於對照組。

表 4-1 程式設計成就測驗 T 檢定結果

	實驗組 (n = 163)		對照組 (n = 160)		t	p	95%CI	
	M	SD	M	SD			LL	UL
程式理解	5.1	1.2	5.2	1.2	-0.716	0.474	-3.65	0.17
程式實作	21.0	3.8	18.0	4.9	5.918	0.000**	1.99	3.97
總分	26.1	4.4	23.2	5.5	5.041	0.005**	1.75	4.01

註：CI = 信賴區間；LL = 下界；UL = 上界。

* $p < 0.05$. ** $p < 0.01$ 。

從下列教學紀錄中，發現實驗組對於專題製作很投入，並能產生自發性學習：分享自己設計的遊戲時，氣氛十分熱烈，看得出每個人臉上充滿期待的表情。（教：3：8-9：w3）

平時一有時間就上網玩遊戲的學生們，竟然出現了「我不玩了，我要繼續作這個（Scratch）」的反應，令人十分驚喜。（教：3：17-18：w3）

「將要完成自己的遊戲」這個念頭對學生有極大的驅動力，明顯可以看出有些平常較吵鬧的學生，上課聽講時安靜許多，中間自行操作的時候，即使提早完成，也會利用機會打開 Scratch 內建的繪圖編輯器，開始繪製自己遊戲會用到的圖案。（教：4：3-6：w4）

有的學生展現出令人感動的熱忱，下課後還會留下來問更多的作法，或是經過自己嘗試，發現一些我之前沒想過的作法。（教：9：20-22：w8）

綜合上述，實驗組在程式設計實作與整體表現上，皆優於對照組，顯示專題導向學習有益於學生學習程式設計，可能的原因在「對專題的投入程度」。實驗組的目標是製作一個「自己（設計）的遊戲」，因此學生對課程都感到興奮而期待，甚至願意將課堂中剩餘的時間用來練習 Scratch。在這樣強烈的驅動下，學生在學習上會較為積極，也願意投入更多時間與心力，在這樣頻繁練習、嘗試與思考之下，因此產生了較好的學習效果。

二、程式設計態度

程式設計態度問卷共 13 題，目的在蒐集實驗組與對照組學生，在「對程式設計的看法」、「解題態度」、「學習動機」、「對程式設計的信心」與「學習興趣」上的顯示是否有差異。依同意题目的程度不同分為五個選項，「非常同意」得 5 分、「同意」得 4 分依此類推，「非常不同意」得 1 分。問卷結果以 T 檢定作分析，發現在對程式設計的看法上，實驗組的平均得分為 12.71（標準差 2.19），對照組的平均得分為 12.03（標準差 2.62），p 值為 0.013，達到顯著差異（ $p < 0.05$ ）；在解題態度上，實驗組的平均得分為 8.40（標準差 1.69），對照組的平均得分為 7.70（標準差 1.80），p 值為 0.000，達到顯著差異（ $p < 0.01$ ）；在信心層面上，實驗組的平均得分為 11.82（標準差 2.58），對照組的平均得分為 11.26（標準差 2.19），p 值為 0.037，達到顯著差異（ $p < 0.05$ ）；在總得分上，實驗組的平均得分 63.34（標準差 9.79），對照組的平均得分為 60.73（標準差 10.16），p 值為 0.005，達到顯著差異（ $p < 0.01$ ）。顯示對於程式設計，實驗組學生比對照組學生有比較正面的看法、積極的解題態度以及較高的信心（如表 4-2）：

表 4-2 程式設計態度問卷之結果分析

	實驗組 (n = 163)		對照組 (n = 160)		t	p	95%CI	
	M	SD	M	SD			LL	UL
1 看法	12.71	2.19	12.03	2.62	2.501	0.013*	0.14	1.2
2 解題態度	8.4	1.69	7.7	1.8	3.61	0.000**	0.19	1.08
3 學習動機	8.42	1.6	8.06	1.85	1.834	0.068	-0.02	0.73
4 學習信心	11.82	2.58	11.26	2.19	2.099	0.037*	0.03	1.08
5 學習興趣	12.4	2.98	11.83	2.88	1.776	0.077	-0.06	1.22
6 總分	53.76	8.83	50.89	9.32	2.844	0.005**	0.42	4.79

註：CI = 信賴區間；LL = 下界；UL = 上界。

* $p < 0.05$. ** $p < 0.01$ 。

以下將以「對程式設計的看法」、「解題態度」、「學習動機」與「學習興趣」四類進行進一步的分析：

（一）對程式設計的看法

從教學紀錄顯示，透過符合其經驗的舉例，讓學生體會「程式設計」是簡單而有意義的，也增進學習的動力：

循序概念以符合學生經驗的「音樂課看譜吹奏笛子」為例，說明當指令積木組合的順序改變時，音符播放的順序也會跟著改變，而當排出某特定順序時，不成調的音符就能譜成曲子。當聽到電腦唱出他們熟悉的曲子，學生們都覺得很有趣，也體驗到原來「指令的順序」是有意義的。（教：3：15-18：w4）

有了互動的功能，離實現自己的遊戲又更近了一步，有的學生已開始想設計的遊戲中，哪一部份可以用這個方法來完成，學生迫不及待地提出一個又一個有關製作遊戲的問題。（教：5：23-25：w5）

「變數」的應用對學生來說不是問題，可能是因為主要作為「時間」和「分數」，這兩個項目對學生來說是很貼近經驗的，所以可以輕易理解。且學生也能提出一些類似的應用，例如：生命值、高度、距離、子彈數量等（教：14：5-8：w13）

根據以上資料，推測實驗組學生認為「程式設計很有趣」的原因，可能是透過專題的製作，讓原本陌生的「程式設計」，與他們的生活經驗有了連結、產生了意義，再加上對「自己設計的遊戲」的期待，因此對程式設計有較正面的看法。

此外，學生能自行發現如何變化程式，以做出更有趣的效果，讓自己的遊戲變得更特別且令人羨慕：

（製作計時效果）有些學生想做出每隔 0.1 秒，將時間變數減少 0.1 的效果（因為這樣顯示出來的計時方式看似比較精準，或以學生的說法是「比較炫」）。（教：13：13-14：w12）

經過分享和觀摩，能促進學生將自己的作品作得更好，以得到更多同儕的羨慕和稱讚。（教：18：21-22：w16）

遊戲測試時的氣氛很好、很愉快，有學生開心的表示，他設計的遊戲吸引了很多同學來玩，或是有人不斷的找他請教「是如何做出某種特殊的效果」。（教：18：24-25：w16）

綜合上述資料，顯示實驗組學生認為完成一段程式能帶來成就感，可能原因在組合一段程式以呈現某些炫麗的遊戲效果，能吸引同儕的讚美與羨慕的眼光，進而獲得成就感，學生的成就感能引發成就動機，促使學童加深學習的興趣，有了學習興趣將對學習動機有正面影響（何偉雲，2001）。

（二）解題態度

「解題態度」旨在了解學生在設計程式遇到困難時，是否願意積極嘗試自行解決。

從教學紀錄內容中，可以發現實驗組對於製作「自己」的專題作品感到有興趣，更有參與感，因此有較高的動力去完成，或願意挑戰更難的設計：

為了「實現」自己設計的遊戲，學生願意不斷地思考、反覆測試自己的程式，比指派固定作業要他們完成，有更高的動力。（教：10：7-9：w9）

在實驗組可以發現，有學生會主動在課後提出……對自己的作品有很強的動機，並希望能夠做出令自己滿意的作品。對照組則有較多的學生在測驗時表現出「兩手一攤便說：我不會做」的態度，即使提供過去練習的檔案與講義參考，也不願花時間去找答案。（教：12：15-19：w11）

綜合上述資料，顯示經過專題導向學習，學生對程式設計的解題態度較為積極、主動，相較於對照組有些學生則是為了應付學期中的測驗，認為無法完成也無所謂，實驗組則是為了「實現」自己設計的遊戲，因此有完成且不斷改進的動力。這樣類似於吳志緯和黃萬居（2003）在國小以電腦樂高進行科學教學活動的發現：於解決問題的歷程中，學生能認知到重要的問題裡還隱藏著許多次要問題，並願意積極面對問題。

（三）學習動機

本研究測量「學習動機」的目的在了解學生學習程式設計的動機是否傾向主動、自發。從教學紀錄中發現，學生會主動提出更豐富的設計，與同學或老師討論：

在實驗組可以發現，有學生會主動在課後提出，他想作的遊戲比較複雜，擔心六節課無法完成，看得出對自己的作品有很強的動機，並希望能夠做出令自己滿意的作品（教：12：15-16：w11）

對照組則有較多的學生在測驗時表現出「兩手一攤便說：我不會作」的態度，即使提供過去練習的檔案與講義參考，也不願花時間去找答案。（教：12：18-19：w11）綜合上述，推測可能原因在於實驗組學生是為了完成自己的作品學習，因此較能自發、主動的練習 Scratch。

（四）學習興趣

「學習興趣」意指學生對學習程式設計的興趣與意願。從教學紀錄中，可以發現學生在程式設計課程中表現出熱忱，討論時氣氛熱絡而愉快：

學生三三兩兩地一邊討論自己設計的遊戲，一邊結伴離開教室，甚至有學生問，除了學習單上設計的這一個，能不能再自己多作一些？（教：2：28-30：w2）

分享自己設計的遊戲時，氣氛十分熱烈，看得出每個人臉上充滿期待的表情。（教：3：8-9：w3）

有的學生展現出令人感動的熱忱，下課後還會留下來問更多的作法，或是經過自己嘗試，發現一些我之前沒想過的作法。（教：9：20-22：w8）

綜合上述，顯示實驗組學生在學習程式設計時有較愉快的學習經驗，推測可能原因是將程式設計結合了貼近學生生活經驗與喜好的內容，增進其學習意願，這個現象類似 Keller（1987a, 1987b）認為的：「先捕捉學習者的注意和興趣，透過經驗的結合，讓學習者體會到所學事物與個人之切身關係，進而產生積極學習意願」。

整體而言，實驗組學生面對程式設計時，比對照組學生有較高的信心、積極的解題態度，對程式設計的看法也比較正面，顯示透過製作遊戲專題的方式學習程式設計，能讓學生對學習程式設計產生信心與正面的看法，並培養積極、主動

的解決問題態度。

三、問題解決能力

(一) 問題解決測驗前測

問題解決測驗前測所得結果以 T 檢定進行分析，如表 4-3。發現實驗組與對照組的平均得分接近，無顯著差異，顯示實驗組與對照組學生在實驗開始前的問題解決能力是相等的。

表 4-3 問題解決測驗前測結果分析

	實驗組 (n = 163)		對照組 (n = 160)		t	p	95%CI	
	M	SD	M	SD			LL	UL
	1 發現問題	5.13	1.47	5.28			1.49	-0.887
2 想出計畫	2.43	1.09	2.51	1.03	-0.685	0.494	-0.32	0.15
3 執行計畫	2.45	0.65	2.58	0.74	-1.596	0.111	-0.28	0.03
4 反省回顧	2.41	1.09	2.62	1.08	-1.662	0.098	-0.45	0.03
5 總分	12.41	2.6	12.99	2.66	-1.925	0.055	-1.17	0.01

註：CI = 信賴區間；LL = 下界；UL = 上界。

* $p < 0.05$. ** $p < 0.01$ 。

(二) 問題解決測驗之前、後測結果比較

實驗結束後，進行問題解決測驗後測（如表 4-4），並以 T 檢定分析實驗組與對照組前、後測的得分差異。本研究呈現問題解決測驗之前後測得分差異結果，如表 4-5。實驗組學生在各方面的得分都高於對照組，其中在「執行計畫」、「反省回顧」以及「總得分」上，p 值分別為 0.038、0.032 以及 0.003，均達到顯著差異。

表 4-4 問題解決測驗後測結果分析

	實驗組 (n = 163)		對照組 (n = 160)		t	p	95%CI	
	M	SD	M	SD			LL	UL
	1 發現問題	5.73	1.68	5.59			1.83	1.26
2 想出計畫	3.03	0.750	2.85	0.896	1.85	0.065	-0.02	0.56
3 執行計畫	2.32	0.763	2.21	0.840	2.08	0.038	0.01	0.47
4 反省回顧	2.61	1.47	2.37	1.63	2.15	0.032	0.04	0.90
5 總分	13.69	3.16	13.01	3.74	3.04	0.003	0.45	2.08

註：CI = 信賴區間；LL = 下界；UL = 上界。

* $p < 0.05$. ** $p < 0.01$ 。

表 4-5 問題解決測驗之前後測結果分析

	實驗組 (n = 163)		對照組 (n = 160)		t	p	95%CI	
	M	SD	M	SD			LL	UL
1 發現問題	0.6	2.04	0.3	2.05	1.27	0.206	-0.16	0.75
2 想出計畫	0.6	1.28	0.33	1.28	1.85	0.065	-0.02	0.56
3 執行計畫	-0.13	0.98	-0.37	1.03	2.09	0.038*	0.01	0.47
4 反省回顧	0.2	1.73	-0.24	1.82	2.15	0.032*	0.04	0.84
5 總分	1.29	3.47	0.19	3.76	3.04	0.003**	0.44	2.08

註：CI = 信賴區間；LL = 下界；UL = 上界。

* $p < 0.05$. ** $p < 0.01$ 。

由上可見，經過專題導向學習，實驗組在「執行計畫」、「反省回顧」方面，以及整體問題解決能力上的表現，有顯著的提升。透過教學紀錄發現學生為了完成自己的遊戲，並使其更完整、更具挑戰性，會更願意去思考、嘗試完成作品；另外，同儕的回饋與稱讚，也會強化學生反思如何修改遊戲的動力：

為了「實現」自己設計的遊戲，學生願意不斷地思考、反覆測試自己的程式，比指派固定作業要他們完成，有更高的動力。（教：10：7-9：w9）

從實驗組製作遊戲的情形觀察發現，「變數」的應用對學生來說不是問題，可能是因為主要作為「時間」和「分數」，這兩個項目對學生來說是很貼近經驗的，所以可以輕易理解。（教：14：5-7：w13）

在努力與投入的意願上而言，實驗組明顯地較對照組高，且主動詢問者多，並願意找尋更多方法加入變化，使自己的作品更有挑戰性。（教：14：13-14：w13）

發現經過第三者的測試跟建議，學生比較會主動修正自己的程式組合。（教：18：16-17：w16）

設計者會發現自己的程式不夠完善，而主動嘗試解決這個問題。可見經過分享和觀摩，能促進學生將自己的作品作得更好，以得到更多同儕的羨慕和稱讚。（教：18：20-22：w16）

然而，在「發現與定義問題」和「想出計畫」方面，與對照組並沒有顯著差異。透過焦點訪談與教學紀錄的分析，歸納出可能原因有二：

1. 課程內容的偏重

從教學紀錄可知，課程一開始是以教學為主，必須讓學生熟悉軟體的功能與環境後，才能利用 Scratch 來製作遊戲專題，時間大約從第二週到第六週。雖有在課堂中引導學生分析即將面對的任務、列出工作清單，並安排進度（第八週），但在製作遊戲專題的過程中，還是以實際操作（如：組合指令積木、檢視執行結

果等)的時間占多數(第八週到第十週)。在這樣的偏重之下,學生有較多的機會,培養執行計畫與反省回顧的能力,卻也壓縮了加強「發現與定義問題」與「想出計畫」的時間,學生對於分析眼前問題、找出已知與未知項目,進而擬定解決計畫的過程都感到「不清楚」、「沒辦法區分」,顯示學生在「發現與定義問題」與「擬訂計畫」階段,沒有獲得有效且足夠的訓練。

2. 教學時間的限制

由於每週只有一節 40 分鐘的課程,不易使學生長時間投入專題之中,使得有些學生對自己的專題進度缺少自覺。另外,課程不夠密集,或遇到全校性活動無法上課的結果,導致學生容易遺忘上一週的課堂內容,需要留一些時間不斷複習,在不熟悉軟體環境與指令積木功能的情況下,需要花更多的心力處理程序性的操作知識,無法全心專注在解決專題上遭遇的問題,產生高層次的思考。

有時學生不知選擇何種指令積木,蠻大一部份原因是不清楚要用的指令積木是哪一類型。(教:10:19-20:w9)

有些學生之前在練習時沒有確實完成,當然也就對如何製作倒數計時不熟悉,也沒有參考檔案可看。(教:15:16-17:w14)

一週只有一節課,學生容易忘記(或說沒有這樣的習慣)要填寫進度表,第一階段遊戲製作完成後,從回收的進度表可看出有的是繳交前趕工而成的。(教:16:2-4:w9)

從上述教學紀錄內容可以得知,一週 40 分鐘的時間,無法讓學生完整沈浸在專題中,也容易遺忘之前的課程。在缺少足夠時間練習的情況下,造成對某些基本操作或功能不熟悉,無法將其自動化處理,便會影響解決專題上遭遇的問題的專注力。

綜合上述資料與分析,實驗結果顯示以專題導向學習法學習程式設計課程,讓學生以製作遊戲為目標,能提升學生學習程式設計的信心,並能培養其嘗試與解決程式設計問題的意願,在學習成就上也有較好的表現。另外,教師在專題進行的過程中,隨時監督學生的學習狀況與進度,適時地提供支持,便能協助學生獲得解決問題的成功經驗,並鼓勵學生將在課程中學習到的問題解決技巧,運用在未來的學習之中。

伍、結論

根據研究結果與討論進行歸納,獲得以下結論:

一、使用專題導向學習的學生,在學習程式設計有較高的學習成就

根據程式設計成就測驗結果發現，實驗組在「程式實作」與程式設計成就測驗總得分上，實驗組學生的得分明顯高於對照組。由教學紀錄推論實驗組成就較高的原因可能在於對專題的投入較多，願意主動練習、嘗試與思考，因此在程式設計上有較好的表現。

二、使用專題導向學習的學生，對學習程式設計有比較正向的看法

根據程式設計態度問卷結果發現，實驗組學生認為學習程式設計「有趣」，並且能在完成一段程式後得到成就感，與對照組學生有顯著差別。另外，實驗組與對照組學生皆認為學習程式設計能讓思考更清晰，這樣的結果與 Shafto (1986) 以及許多其他研究者 (Barker, Lao, Reynolds, & Wu, 2008) 認為學習程式設計可以幫助思考、組織，以清楚、正確的方式描述如何解題的看法相符。

三、使用專題導向學習的學生，有比較積極的程式設計解題態度

根據程式設計態度問卷結果發現，實驗組學生在程式設計上遭遇問題時，「願意自己嘗試解決」並「喜歡經過自己嘗試完成一段程式的感覺」，顯示使用專題導向學習的學生，在程式設計解題上，較使用傳統方式學習的學生有更積極的態度。

四、使用專題導向學習的學生，有比較好的「執行計畫」能力

根據問題解決測驗結果發現，使用專題導向學習的學生，比使用傳統方式學習的學生，在「執行計畫」方面具備較好的問題解決能力。推測可能原因在於實驗組是為了實現自己設計的遊戲，因此願意不斷的嘗試、思考，尋求更好的解決方法，因此在「執行計畫」方面，有較好的表現。

五、使用專題導向學習的學生，有比較好的「反省回顧」能力

根據問題解決測驗結果發現，使用專題導向學習的學生，比使用傳統方式學習的學生，在「反省回顧」方面具備較好的問題解決能力。推測可能原因在於實驗組是為了讓自己設計的遊戲更完美，且希望得到同儕的羨慕與讚美，因此願意反覆修改程式，挑戰更難的遊戲設計，故於「反省回顧」方面，有較好的表現。

參考文獻

- 何偉雲 (2001)。初步探討影響學童自然科學習成就因素的排序。屏東師院學報，14, 933-952。
- 吳志緯、黃萬居 (2003)。以電腦樂高進行國小科學教學活動設計與實施之研究。科學教育研究與發展季刊 2003 專刊，53-87。
- 吳正己、何榮桂 (1998)。高級中學新訂電腦課程的內涵與特色。科學教育月刊，



- 208, 26-32。
- 吳正已、林凱胤(1997)。問題解決導向的程式語言教學。資訊教育雜誌創刊十年特刊, 75-83。
- 邱貴發(1996)。情境學習理念與電腦輔助學習——學習社群理念探討。臺北市: 師大書苑。
- 潘怡吟(2002)。遊戲型態教學對國小學生自然與生活科技學習之研究。未出版之碩士論文, 臺北市立師範學院科學教育研究所, 臺北市。
- Ahlers, R., Driskell, J., & Garris, R.(2002). Games, motivation, and learning: A research and practice model. *Simulation and Gaming*, 33, 441-467.
- Barker, G., Lao, A. C., Reynolds, B. L., & Wu, F.(2008). *Learning Efficiencies Using Multi-Agent Based Game Simulations*. Paper presented at the ICCE, Taipei, Taiwan. Abstract retrieved from <http://www.apsce.net/icce2008/papers/ICCE2008-paper70.pdf>
- Bishop, C. C., Courte, J., Evans, D., & Howard, E. V.(2007). A Quantitative and Qualitative Investigation of Using Alice Programming to Improve Confidence, Enjoyment and Achievement Among Non-Majors. *Journal of Educational Computing Research*, 37(2), 193-207.
- Bouhuijs, P.A.J., Perrenet, J.C., & Smits, J.G.M.M.(2000). The Suitability of Problem-based Learning for Engineering Education: *Theory And Practice. Teaching in Higher Education*, 5(3), 345-358.
- Brown, Q., Mongan, W., Garbarine, E., Kusic, D., Fromm, E., & Fontecchio, A.(2008, June). *Computer aided instruction as a vehicle for problem solving: Scratch programming environment in the middle years classroom*. Proceeding of American Society for Engineering Education 2008. Pittsburgh, PA.
- Copper, S., Dann, W., & Pausch, P.(2003). Using Animated 3D Graphics to Prepare Novices for CS1. *Computer Science Education*, 13(1), 3-30.
- Chao, P. Y.(2016). Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Education*, 95, 202-215.
- Dyck, J. L., Mayer, R. E., & Vilberg, W.(1986, July). Learning to program and learning to think: what's the connection? *Communication of the ACM*, 29(7), 605-610.
- Hadgraft, R., Ilic, V., & Scott, N.(2003). Engineering education-is problem-based or project-based learning the answer? *Australasian Journal of Engineering Education*, 3, 2-14.
- Hayes, E.(2008). Game content creation and it proficiency an exploratory study. *Computers & Education*, 51, 97-108.



- Idlbi, A.(2009). Taking Kids into Programming(Contests)with Scratch. *Olympiads in Informatics*, 3, 17-25.
- Keller, J. M.(1987a). Strategies for stimulating the motivation to learn. *Perfomance and Instruction*, 26(8), 1-7.
- Keller, J. M.(1987b). The systematic process of motivational design. *Perfomance and Instruction*, 26(9), 1-8.
- Kiili, K.(2007). Foundation for problem-based gaming. *British Journal of Educational Technology*, 38, 394-404.
- Lin, M., & Pedersen, S.(2003). Teachers' beliefs about issues in the implementation of a student-centered learning environment. *Educational Technology, Research and Development*, 51(2), 57-76.
- Lin, J. W., & Tsai, C. W.(2016). The impact of an online project-based learning environment with group awareness support on students with different self-regulation levels: An extended-period experiment.*Computers & Education*, 99, 28-38.
- Mayer, R. E., & Wittrock, R. C.(2006). Problem solving. In P. A. Alexander & P. H. Winne(Eds.), *Handbook of educational psychology*(pp.287-304). Mahwah, NJ: Erlbaum.
- Mills, J.E., & Treagust, D.F.(2003). "Engineering Education—Is Problem-Based or Project-Based Learning the Answer? *Australasian Journal of Engineering Education*. Retrieved from http://www.aae.com.au/journal/2003/mills_treagust03.pdf
- Salomon, G., & Perkins, D. N.(1987). Transfer of cognitive skills from programming: When and how? *Journal of Educational Computing Research*, 3, 149-170.
- Seidman, R. H.(1988). New directions in educational computing research. In R. E. Mayer(Eds.). *Teaching and learning computer programming: Multiple research perspectives* (pp. 299-308). Hillsdale, NJ: Lawrence Erlbaum.
- Shafto, S. A. S.(1986). Programming for learning in mathematics and science. *ACM SIGCSE Bulletin*, 18(1), 296-302.
- Shaw, D. G.(1986). Effects of learning to program a computer in BASIC or Logo on problem-solving abilities. *AEDS Journal*, 19, 176-189.
- Tanweer, M. R., Suresh, S., & Sundararajan, N.(2016). Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems. *Information Sciences*, 326, 1-24.
- Thomas, J. W.(2000). *A review of research on project-based learning*. Retrieved from http://www.bie.org/index.php/site/RE/pbl_research/29
- Wing, J.(2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35.



Graduate School of Educational Communica- tions and Technology College of Science

Wen-Hui Chang

Ph.D. candidate

National Taipei University of Education

E-mail: 0805@gmail.com

Chia-Ti Wu

Teacher

Juguang Elementary School

E-mail : charity1113s@gmail.com

Tung-Hsuan Chen

Ph.D. candidate

National Taipei University of Education

Abstract

The purpose of this study is to increase the ability of the students' problem solving and their knowledge application with a sixteen-week programming course. The participants were 323 fourth graders from an elementary school. The experimental group received project-based instruction while the control group received traditional instruction. The data collected consist of quantitative and qualitative data. The quantitative data were analyzed by Chi-square test and T-test, including results of pre-test and post-test on problem-solving, questionnaire of attitude toward programming, and achievement test of programming. The qualitative data include teaching records and interview records. After the sixteen-week instruction, comparing to the control group, project-based learning benefits the experimental groups in (1) programming learning, (2) holding positive perspectives of programming, (3) solving problems actively, (4) building better abilities of executing plans, and (5) acquiring better abilities of reflection and retrieving. The present study indicates that project-based learning in programming course assists students in programming learning as well as problem-solving abilities. Further-



more, it is suggested that teachers monitor the students' progress, guide them to confront the problems and seek solution in order to help them to transfer the skills and experiences they have learnt from the course in their daily life.

Key words: Project-based Learning, Scratch programming software, Programming, Problem-solving