

國立交通大學

理學院科技與數位學習學程

碩士論文

完成問題策略對基本程式概念教學的學習成效研究
——以國小四年級學童為例

The Study of Learning Achievements for the Completion Example
Strategy on Basic Programming Concepts - Taking Forth Grade
Elementary School Students as an Example

研究生：洪詩玲

指導教授：陳登吉 教授

中華民國九十九年六月

完成問題策略對基本程式概念教學的學習成效研究
—以國小四年級學童為例

The Study of Learning Achievements for the Completion Example
Strategy on Basic Programming Concepts—Taking Forth Grade
Elementary School Students as an Example

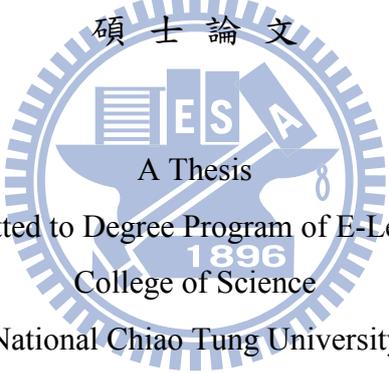
研究生：洪詩玲

Student：Shih-Ling Hung

指導教授：陳登吉

Advisor：Deng-Jyi Chen

國立交通大學
理學院科技與數位學習學程
碩士論文



A Thesis
Submitted to Degree Program of E-Learning
College of Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Degree Program of E-Learning

June 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

完成問題策略對基本程式概念教學的學習成效研究

—以國小四年級學生為例

學生：洪詩玲

指導教授：陳登吉 教授

國立交通大學理學院科技與數位學習學程

摘要

目前國內中小學資訊教育多著重在應用軟體的學習，雖在日常生活中有其需要性，但國小學生正值各項能力蓬勃發展時期，反覆教導學生應用軟體的操作不免有些可惜。而兒童透過程式設計可以提升高層次的思考能力，像問題解決、邏輯思考能力等，此種能力與教育部（2008）中所提及到中小學學生所應具備的十大基本能力「獨立思考與問題解決」正可相互連結。

美國計算機協會（Association for Computing Machinery, ACM）2003 年為國中小資訊教育所發展的課程模式中指出，國小到國中二年級的學生除了要學習基礎的電腦科學，也應整合演算法思考（algorithmic thinking）來解決日常生活中的問題，而程式設計即為培養此種能力的最佳途徑。在資訊普及的社會中，具備讀寫算的基本能力已不敷使用，應將程式設計視為通識教育一般，向下紮根導入到國小教育，讓我們的學生更具備與國際接軌的能力。

但程式設計包含許多複雜的思考歷程，學習者除了要面對陌生的程式語法、指令及程式語言軟體介面，更為困難的是要將腦海所規劃的問題解決方案轉化為具體的程式指令，此種學習歷程常常會造成學習者過多不必要的認知負荷。

因此，引發研究者的研究動機，身為一線的資訊教師在教導兒童學習程式設計時，教學方法應與一般應用軟體教學有所不同，教師必須規劃適當的教學策略來引導學生學習，以期達到較好的學習成效。

本研究採準實驗研究法，以臺北市某國小四年級學生為研究對象，共 91 人進行分組實驗，歷時十一週。研究目的為探討教師以不同的教學策略教導國小四年級學生學習基本程式概念對其學習成效上的影響。所使用的教學策略為一般程式教學、問題導向教

學與完成問題教學。教學內容為控制結構 (control structure)，分成順序平行、重複執行與條件執行三個子單元，以成就測驗為測量工具，分為各單元形成性測驗及總結性測驗，而測驗類型上均包含紙筆測驗與程式實作。測驗時間為各單元結束後進行形成性測驗，並於教學最後一週進行總結性測驗及填寫學習態度問卷。

實驗數據經統計分析，研究結果如下：

1. 在不同的教學策略下，學生整體學習成效，完成問題組優於問題導向組，且達顯著水準。
2. 在不同的教學策略下，學生在各單元形成性測驗，只有條件執行單元的學習成效有顯著差異，且以完成問題組學習成效優於問題導向組。
3. 在不同教學策略下，針對紙筆測驗成績，學生在重複執行與條件執行單元上，完成問題組優於問題導向組，且達顯著水準。
4. 在不同教學策略下，針對程式實作成績，學生在順序平行與遊戲實作上，完成問題組優於問題導向組，且達顯著水準。
5. 以不同電腦基本能力將學生分群進行統計分析。低能力組別的學生，在不同的教學策略下其學習成效有顯著差異，且以完成問題組優於問題導向組。

關鍵字：完成問題策略、問題導向策略、基本程式概念、程式設計

The Study of Learning Achievements for the Completion Example Strategy on
Basic Programming Concepts--Taking Forth Grade Elementary School
Students as an Example.

student : Shih-Ling Hung

Advisors : Dr. Deng-Jyi Chen

Degree Program of E-Learning
College of Science
National Chiao Tung University

Abstract

About the present information education in Taiwan, it focuses on how to operate application softwares although it do help, but it seems to be not enough, especially it is during the period that elementary school students dig out their potentials. Programming could improve children's thinking skills such as "problem solving abilities" and "logical reasoning"; "think indepently and problem solving" is in the top ten basic abilities announced by the Ministry of Education elementary and junior school students shall have, and programming is one of the methods to reach this goal.

In the US, there is a suggestion indicated in the 2003 computer education courses for elementary and junior school students announced by ACM (Association for Computing Machinery) that except for basic computer science, students should learn about "algorithmic thinking" and solve problems through "conditional" and "repetitions".

In other words, it will let students fall behind future world of they are only equipped with Read/Write/Calculate abilities. On the contrary, it is necessary to regard programming as one part of general education of elementary school period.

Programming includes a lot of complicated thinking process and implementation. For an beginner, except for getting familiar with syntax, command, and programming environments, it would be more difficult to tansfer the problem solving methods into exact programming coding and cause high cognitive load. Therefore, the reason to do this research is to help computing teachers to come up with some good ideas for teaching programming courses.

This research is executed by "quasi-experimental approach" and take three forth grade

elementary school classes (total 91 students) in Taipei as subjects. After experimental teaching for 11 weeks, evaluation tests about learning achievement are carried out. The course is about "control structures", and the instructional strategies are "traditional teaching", "problem-based teaching", and "completion example teaching". The main purpose of this research is to probe into the learning achievement of the fourth grade elementary school students about basic programming concepts by using different instructional strategies.

The experimental results come up from statistical analysis are:

1. It reaches level of significance that the overall learning achievement is better by using "completion example teaching" strategy compared with "problem-based teaching" strategy.
2. Through tests for different subjects, it is only significant that the learning achievement is better by using "completion example teaching" strategy compared with "problem-based teaching" strategy in condition statement subject.
3. It reaches level of significance that the learning achievement is better by using "completion example teaching" strategy compared with "problem-based teaching" strategy through paper tests on repetition & conditioning implementation.
4. It reaches level of significance that the overall learning achievement is better by using "completion example teaching" strategy compared with "problem-based teaching" strategy through programming exercises of sequence and parallel and game design.
5. To classify students into different levels of computing ability, the learning achievement is much better by using "completion example teaching" strategy compared with "problem-based teaching" strategy.

Keywords: completion example strategy, problem-based strategy, basic programming concept, programming design

誌謝

光陰似箭，轉眼間兩年的研究生涯也已近尾聲，回想起當初進入研究所的初衷，也不過是秉持著學習進修的心情，想要在工作外再次享受當學生的幸福。在交大求學的兩年間有歡笑也有淚水，歡笑的是跟著同學們一起學習一起成長；淚水的是兩年期間每週一次的往返台北—新竹，隔天還要上班的辛苦，而走到現在回想起來，一切都是值得的。

在撰寫論文的漫漫長路上，首要感謝的是指導教授陳登吉教授，與教授的對談間，常有「聽君一席話，勝讀十年書」的感覺，每當與教授對談過後，總覺得獲益良多。以及孔崇旭教授每週不辭辛勞的往返台中—新竹，除了在論文上給予細心的指導與建議，讓研究能順利完成，也從教授身上學習到處理事情的方法。另外，班導師李榮耀教授也多次於論文研討課程上給予指導，口委賴阿福教授、黃世昆教授在論文口試時也不吝給予學生許多指導與建議，讓此份論文更趨完善。

而最艱苦的研二時期，一直陪伴著我的實驗室夥伴淑芬、元鴻、錦儒和建福，以及同學惠晴、佩娟，大家相互勉勵打氣、一起努力，都是自己可以繼續往前的最大動力。也很感謝多媒體實驗室的學長姐及學弟妹們，雖然不是同一屆或是同一科系的但是對於在職專班的我們總是諸多照顧及幫忙。

撰寫論文的這段期間，默默幫助我最多的還是家人和朋友，最大的感謝要給予我的父母，感謝阿母瘋狂不計成本的燉煮雞湯幫我補身體，讓我可以課業與工作兩頭燒的時候還有足夠的精神撰寫論文，每逢週三晚睡為我等門的阿爸，還有一直幫忙做家事不想給我添負擔的大姐、一直給予我鼓勵的二姐、陳年好友秀娟與美瑩、默默忍受孤獨沒有人陪的男友大毛，有你們在我背後默默的幫助我，當我感到疲憊、無助的時候，想到你們就會有動力繼續努力。

而在這期間也遇到許多幫助我的貴人，有學長柏豪總是給我很棒的意見、在工作上挺我的均地組長、不吝給予我建議的保煌老師還有交大北大門的土地公公，多虧有你們助我一臂之力，此份論文才得以完成。

在交通大學這兩年，十足成長了許多，不論是在專業知識上還是心智上，最後感謝教導我許多知識的所有專班任教老師們，很感謝我身邊的所有人給予我的支持與協助，謝謝你們，謹以本論文獻給所有幫助我、支持我、關心我的人。

洪詩玲謹誌

中華民國九十九年六月

目 錄

摘要	i
Abstract	iii
誌 謝	v
目 錄	VI
表目錄	VIII
圖目錄	X
一、緒論	1
1.1 研究背景	1
1.2 研究動機	3
1.3 研究目的及問題	4
1.4 名詞解釋	4
1.4.1 完成問題策略 (Completion Example Strategy)	4
1.4.2 問題導向策略 (Problem-based Strategy)	5
1.4.3 基本程式概念 (Basic Programming Concepts)	5
1.4.4 學習成效 (Learning Achievement)	5
1.4.5 學習態度 (Learning Attitude)	5
1.5 研究範圍與限制	6
1.5.1 研究範圍	6
1.5.2 研究限制	6
二、文獻探討	7
2.1 兒童程式設計教學	7
2.2 認知負荷	9
2.3 完成問題教學策略	12
2.3.1 相關文獻探討	12
2.3.2 完成問題教材設計	14
2.4 問題導向教學策略	14
三、完成問題策略教材設計理論模型	15
3.1 模型說明	15
3.2 實際應用	17
3.2.1 追蹤觀察	17
3.2.2 模仿修改	18
3.2.3 擴充創作	19
四、研究方法	20
4.1 研究流程	20

4.1.1 準備階段	20
4.1.2 實驗階段	21
4.1.3 資料分析階段	21
4.2 研究設計	22
4.2.1 研究變項	22
4.3 研究對象	24
4.4 研究工具	25
4.4.1 教學實驗相關教材	25
4.4.2 基本程式概念成就測驗	34
4.4.3 學習態度問卷	36
4.5 實驗程序	37
4.6 資料處理與分析	39
五、結果與討論	40
5.1 學習成效結果分析	40
5.1.1 各組總結性測驗學習成效分析	40
5.1.2 各組形成性測驗學習成效分析	41
5.1.3 各組紙筆測驗學習成效分析	44
5.1.4 各組程式實作學習成效分析	46
5.1.5 高能力組中，不同教學策略對基本程式概念學習成效之分析	48
5.1.6 中能力組中，不同教學策略對基本程式概念學習成效之分析	48
5.1.7 低能力組中，不同教學策略對基本程式概念學習成效之分析	49
5.1.8 小結	50
5.2 學生學習態度問卷分析結果	51
5.2.1 學生情意態度	51
5.2.2 針對學生學習動機進行探討	52
5.2.3 針對各單元難易度進行探討	52
5.2.4 不同策略對學生學習上的幫助程度	53
六、結論與建議	55
6.1 結論	55
6.2 未來研究方向	57
參考文獻	59
附錄	64

表目錄

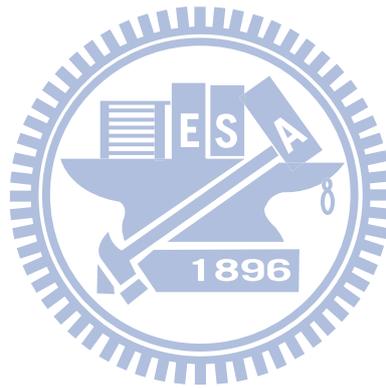
表 1 實驗各組人數分配表	24
表 2 各組上學期電腦科平均成績變異數分析	24
表 3 教學單元與教材分析表	25
表 4 教學單元程式概念與Scratch指令範例表 (續)	27
表 5 追蹤觀察活動—條件執行單元為例 (續)	30
表 6 模仿修改活動—條件執行單元為例	32
表 7 擴充創作活動—條件執行單元為例	32
表 8 測驗題目分配雙向細目表	34
表 9 測驗題目類型與配分	35
表 10 測驗試題難度與鑑別度	35
表 11 學生學習態度問卷向度及題目分配表	36
表 12 教學實驗週次表	38
表 13 數據分析細項表	39
表 14 各組總結性測驗成績描述性統計量	40
表 15 各組總結性測驗成績變異數分析摘要	41
表 16 各組總結性測驗成績事後比較分析摘要	41
表 17 各組在順序平行單元成績描述性統計量	41
表 18 各組在順序平行單元成績變異數分析摘要	42
表 19 各組在重複執行單元成績描述性統計量	42
表 20 各組在重複執行單元成績變異數分析摘要	42
表 21 各組在條件執行單元成績描述性統計量	42
表 22 各組「條件執行單元成績變異數分析摘要	43
表 23 各組在條件執行單元學習成績進行事後比較分析摘要	43
表 24 各組在紙筆測驗上的描述性統計量	44
表 25 各組紙筆測驗變異數分析摘要總表 (續)	44
表 26 各組在重複執行及條件執行紙筆測驗上事後比較分析摘要總表	45
表 27 各組程式實作的描述性統計量	46
表 28 各組程式實作測驗上的變異數分析摘要總表 (續)	46
表 29 各組在順序平行及遊戲程式實作事後比較分析摘要總表	47
表 30 高能力組學生在總結性測驗描述性統計量	48
表 31 高能力組中，不同教學策略在總結性測驗成績變異數分析摘要	48
表 32 中能力組在總結性測驗成績描述性統計量	48
表 33 中能力組中，不同教學策略在總結性測驗成績變異數分析摘要	49
表 34 低能力組學生在總結性測驗成績描述性統計量	49
表 35 低能力組中，不同教學策略總結性測驗成績變異數分析摘要	49
表 36 低能力組中，不同教學策略在總結性測驗成績的事後分析比較表	49

表 37 學習成效分析總表	50
表 38 不同能力分組學生的學習成效分析總表	50
表 39 學生對程式設計課程是否覺得有趣調查結果	51
表 40 學生對學習程式設計課程是否感到愉悅調查結果一	51
表 41 學生對學習程式設計課程是否感到愉悅調查結果二	51
表 42 學生學習動機調查結果一	52
表 43 學生學習動機調查結果二	52
表 44 順序平行單元難易度調查結果	52
表 45 重複執行單元難易度調查結果	52
表 46 條件執行單元難易度調查結果	52
表 47 問題導向策略對學生學習上的幫助程度	53
表 48 一般教學策略對學生學習上的幫助程度	53
表 49 完成問題教學策略對學生學習上的幫助程度	54



圖目錄

圖 1 完成問題策略教材設計理論模型	16
圖 2 追蹤觀察範例說明	17
圖 3 追蹤觀察學習單範例	18
圖 4 模仿修改活動	19
圖 5 擴充創作活動	19
圖 6 研究流程圖	21
圖 7 實驗設計架構圖	22
圖 8 電流急急棒遊戲與控制結構程式概念結合圖	26
圖 9 Scratch 1.4 版 界面說明	27
圖 10 一般程式組教學流程圖（兩堂課）	28
圖 11 完成問題組教學流程圖（兩堂課）	29
圖 12 問題導向組教學流程圖（兩堂課）	33
圖 13 實驗程序流程圖	37



一、緒論

1.1 研究背景

為了能夠因應複雜以及科技化社會所產生的各式各樣問題，運用電腦化的思維（computational thinking）來解決問題已經是每個人均必需具備的基本技能，而電腦化的思維指的即是運用與電腦相仿的思考方式有邏輯、有系統的解決問題，而程式設計則是發展電腦化思維最好的方式（Sutner, 2003; Orr, 2009）。

而目前台灣的資訊教育（information education）中，針對國小學生的程式教學進行探討的研究者，多數將程式設計作為一種心智開發的工具，讓兒童藉此提升高層次的思考能力。像是楊書銘（2008）利用 Scratch 程式語言來增強兒童的邏輯推理能力、問題解決能力及創造力。王麒富（2009）則利用行動研究法，探討 Scratch 對兒童問題解決能力提升之效益研究。而賴建二（2003）發展一套兒童視覺化中文程式語言—小小程式設計師，其立意也是認為兒童學習程式設計，對於問題解決、邏輯思考能力有所助益，所以發展這套中文視覺化的程式設計軟體以降低兒童學習程式語言的「語言」門檻。

若以學科的角度來看程式教學，其所能為兒童增加的能力，與 97 年國民中小學九年一貫課程綱要於十大基本能力中，明列學生應培養「獨立思考與解決問題」的現代國民基本能力是不謀而合的。

學習程式設計除了可以為兒童帶來上述的能力外，學習程式設計本身就是一種最好的學習，Valente（1995）指出在學習程式設計時，其過程是不斷的進行「描述-執行-反思-除錯」的步驟，這是一個不斷循環的過程，而這樣的學習方式，對兒童來說是寶貴的學習訓練。

目前國小學生的資訊能力在正規的課程中，是利用一週一節的資訊課來培養，實際課程的現況多以軟體的操作應用為主（蔡依玲，2007），像是中英打字練習、文書處理、作業系統基本操作、影像處理等，這樣的一個學習內容雖說有其必要性，但國小學生正值各種能力在蓬勃發展的重要時期，在此階段反覆的學習如何操作應用軟體，不免有些可惜（林美娟，2008）。再者，對照我國資訊教育課程的內涵及目標，其中像是「應用分析」、「創造思考」、「問題解決」等能力（國民教育司，2009；張琬翔，2008），也是只教導兒童操作應用軟體所難達到的（林美娟，2008）。

但反觀國外，美國計算機協會（Association for Computing Machinery, ACM）2003 年為州教育廳（state department of education），針對 K-12 的資訊課程規範出一個課程框架，目前最新為「國中小電腦科學模型課程—第二版本」(The ACM Model Curriculum for K-12 Computer Science, 2nd Edition)。在其課程內容中將電腦科學的教學內容分為四個階

段，並分別提出相關的課程建議。第一階段針對K-8 提出建議，第二階段為九年級或十年級，而第三、四階段則是繼續往上發展。在第一階段指出應讓國小學生學習電腦科學的基礎 (foundations of computer science)，並整合簡單的演算法思考 (algorithmic thinking)，更加以具體化來說，即是透過簡單的條件 (if)、迴圈 (loops) 來解決日常生活中的問題。

相較之下，國內資訊教育對於程式設計的學習是從七年級開始，所學的內容從認識程式語言到瞭解其功能與應用，而真正程式設計工具的使用、邏輯演算以及除錯偵測則是到高中才會進行教學。

97 年國民中小學九年一貫課程綱要 (100 學年度實施) 開宗明義指出，培養每個國民具備運用資訊科技的基本知識與技能，此為世界各國資訊教育發展的共同趨勢，而只具備讀、寫、算的基本素養已在目前社會中不敷使用，應將資訊科技能力列為每個國民應具備的基本素養。而此種概念也與 Hartmann、Nievergelt 與 Reichert (2001) 的看法不謀而合，Hartmann 等人指出在通識教育中，應將程式設計 (programming) 的學習視為與讀寫算一樣，成為學生的基本能力之一，並特別強調雖然應用軟體的學習也是資訊科技的學習，但是程式設計的學習可以讓學生清楚的瞭解電腦運作的原理，這種能力才足以應付未來應用軟體的改變。

從 1960 年代 Papert 開發出一套適合兒童學習的 logo 程式語言。開發至今，適合初學者學習的程式語言工具蓬勃發展，兒童學習程式設計已非無法實現的目標。Kelleher 和 Pausch (2005) 更將這些大量的程式語言與程式設計環境做了詳盡的整理，其中有多種工具適合國中小學使用，像 Alice、MSWLogo 以及 Stagecast Creator 等，所以在程式語言工具上的挑選已是不虞匱乏。而 2007 年由麻省理工學院所發展出的 Scratch，跟一般傳統的程式語言相較之下，少了晦澀艱深的語法，減低學生學習的障礙，讓兒童得以接觸程式設計，加上其多媒體的特色，更可引發學生的學習興趣 (Resnick, 2007)。Fain (2004) 更說到兒童在國小四、五年級就已具備抽象推理能力，只要教師設計生動、有趣的教材，讓兒童在學習時可以感受到愉悅以及成就感，即使是小學生也可以學習「真正的」程式設計。

所以在目前的國小資訊教育中應適當的導入基本程式教學 (林美娟, 2008; 楊書銘, 2009; 賴建二, 2003)，甚至在兒童四年級時就應該開始利用程式設計培養其從實作中學習的精神 (黃文聖, 2000; 徐龍政, 1994)。

1.2 研究動機

陳明溥（2007）歸納指出許多程式設計教學研究，發現學生學習成效不佳的原因來自於，教師的教學方法不當以及學習者的個別差異，這兩個因素導致學習者沒有辦法有效運用程式設計的知識於問題解決上。而劉明洲、陸錫峰及林鴻龍（2003）也說明，雖然針對學習者的特質進行教學上的改變是有其意義的，也符合建構教學的理念，但學習者的個人特質很多是教師無法改變的，像是性別、學習型態等，而從教學上進行改進則是較積極的作法。

學者 Harel（1991）使用不同的教學方法，教導四年級學生利用 Logo 學習分數，其控制組是傳統教學，實驗組使用 CAI（Computer-assisted Instruction）教學，實驗結果顯示，實驗組分數顯著高於控制組。因此 Harel 指出教師在教導兒童學習程式設計時，除了要考慮到學習者的特性以外，教師所使用的教學方法更是可以影響學生學習結果的重要因素。

一般人皆認為程式設計是困難的（Boulay, 1986; Scholtz & Wiedenbeck, 1992; Garner, 2009），因為程式設計為一複雜性的思考歷程，在這其中牽扯到許多認知技能及其他相關因素（賴建二，2003；尹致君，1991）。Smith、Cypher 與 Tesler（2000）指出對一開始學習程式設計的人來說，最常遇到的問題就是不知如何將心中規劃的問題解決方案，轉化為程式語言。

總合來說，程式設計的過程牽涉到複雜的認知歷程，學習者在學習過程當中除了需要瞭解程式的語法以及指令，還需要具備問題解決技巧，而 Govender 與 Grayson（2006）更指出最主要影響到學習者的程式能力就是問題解決的技巧。

而就廣義來說，程式設計不只是一堆指令的組合，它所帶代表就是一連串的問題解決步驟（陳明溥，2007；譚俊彥，1993；Winslow, 1996），因此當學習者一開始面臨到問題的情境，除了要瞭解問題，找出解決方案，還要學習程式新知，這對學生來說會造成沉重的認知負荷導致學習成效不佳。

結合上述原因，引發出研究者的研究動機，除了要瞭解學生在學習此種領域的學習困難之外，也應積極的針對教師的教學方法進行探討與設計，用正確有效率的方法來教導學生學習程式設計。而程式設計領域本身的特性是教師很難去改變的，只有透過設計好的教學策略，降低學生的學習門檻。因此在閱讀相關文獻之後，國內目前針對國小程式設計教學的研究雖然不多，且多以問題導向（林裕雲，2002；陳明溥，2007；黃文聖，2000；楊書銘，2008）為主要的教學策略。因此研究中欲探討以不同的教學策略進行基本程式概念教學對學生學習成效的影響。期能以此研究結果，給予一線的資訊教師在教導程式設計課程時的相關參考。

1.3 研究目的及問題

基於上述背景與動機，本研究的目的主要在探討不同的教學策略對學生在基本程式概念學習成效的影響。以期可提供給國小資訊教師在設計程式教學活動時的相關參考。

主要研究目的有以下三點：

- 一、探討「問題導向」、「一般程式教學」及「完成問題」三種教學策略，對國小四年級學生在學習基本程式概念上學習成效的影響。
- 二、針對不同電腦基本能力的學生（高能力組、中能力組、低能力組），探討不同的教學策略對其在學習基本程式概念上學習成效的影響。
- 三、探討以不同的教學策略進行教學，學生學習態度上的差異。

針對研究目的，提出待答問題如下：

- 一、學生基本程式概念的學習成效是否因不同的教學策略而有所差異？
- 二、針對不同的學習單元，分別運用哪種策略其學習成效較佳？
- 三、針對高能力組的學生，以不同教學策略進行教學，其學習成效上是否有所差異？
- 四、針對中能力組的學生，以不同教學策略進行教學，其學習成效上是否有所差異？
- 五、針對低能力組的學生，以不同教學策略進行教學，其學習成效上是否有所差異？
- 六、在不同的教學策略下，學生的學習態度為何？

1.4 名詞解釋

1.4.1 完成問題策略（Completion Example Strategy）

範例（worked example）是一種教學工具，其中包含著問題解決的模型，主要被使用來教導問題解決（problem solving）的技巧。範例的內容包含著詳細的解題步驟以及問題的最後答案（Gruber, Mandl, Renkl, & Stark, 1998）。

而完成問題（completion example）為範例過渡到練習題（practice）之間所產生的混合型態。在一個完成問題中，只展示部分的範例給學生閱讀，而剩下的部份則當成是練習留給學生完成（Clark, Nguyen, & Sweller, 2005）。

本研究中，運用完成問題的概念導入到實際教學策略，詳細教學步驟如下：

課程一開始進行觀察「範例」，學習者透過此步驟觀察到教師要教導的抽象概念，接著再進行「完成問題」的活動，讓學習者透過閱讀教師給予的部份程式並完成剩下的程式區塊，最後學習者仍是要獨立完成一個任務。策略活動主要分為「追蹤觀察」、「模仿修改」、「擴充創作」三部分。

1.4.2 問題導向策略 (Problem-based Strategy)

問題導向的學習方法強調學生的學習活動，與傳統以教師講述為中心不同，學生要自己設定學習目標以及運用搜尋資訊的技巧來解決問題 (Nuutila, To'rma, & Malmi, 2005)。而在程式設計領域中，綜合歸納鍾靜宜 (2003) 與楊書銘 (2008) 的問題解決步驟，發展出本研究中所使用的問題導向教學策略，詳細教學活動分為「瞭解問題」、「分析問題」、「撰寫程式」、「觀察與運作程式」四步驟，最終的問題解決主要以完成電流急急棒遊戲為最後目的。

1.4.3 基本程式概念 (Basic Programming Concepts)

ACM (2003) 提到 k-8 階段的學生應學會演算的思考，而演算法 (algorithm) 簡單來說即是一步步解決問題的方法，當學生遇到特殊問題時，也要會運用條件或是迴圈來找出解決的方法。對照到程式設計上，在演算法中用以決定處理程序應以何種順序執行的結構稱為控制結構 (control structure)，而條件與迴圈則為控制結構的一環。

在本研究中的教學內容，目前國小現階段並無程式設計課程可供參考，因此將基本程式概念的內涵設定為基本的控制結構的學習。總共分為三個教學單元，分別為：

(1) 順序平行 (sequence and parallel) 單元 (2) 重複執行 (looping) 單元以及 (3) 條件執行 (condition statement) 單元。

1.4.4 學習成效 (Learning Achievement)

本研究中的學習成效即為學生在成就測驗 (achievement test) 上的表現，成就測驗在此次教學實驗中依時間點分為形成性測驗 (formative test) 以及總結性測驗 (summative test)。在題型上分成紙筆測驗與程式實作兩個項目，兩項成績總合得分愈高，表示學生在基本程式概念的學習成效愈佳。

1.4.5 學習態度 (Learning Attitude)

本研究的學習態度包含學生對於此次教學課程的感受及看法。實際內涵包括情意態度、學習動機以及教師在使用不同策略教學時，學生在實際學習上所感受到的幫助程度。

1.5 研究範圍與限制

1.5.1 研究範圍

本研究之教學教材為研究者自編教材，教學內容以基本程式概念—控制結構為主，研究主旨為探討教師利用不同的教學策略針對基本程式概念進行教學，探討對不同組別學習成效的影響，教學對象為國小四年級學生。期盼此研究結果能提供給其他資訊教師在教導程式設計相關課程時，作為教學設計的參考。除上述以外之其他因素則不在本研究探討範圍。

1.5.2 研究限制

一、研究樣本的限制

本研究之教學實驗對象，為臺北市某國小四年級學生，其他地區與其他年級的學生不在此次研究範圍內。

二、研究工具的限制

研究所使用的教學軟體採用 Scratch 程式設計軟體，此套軟體利用圖像式的方式來撰寫程式，而其相關程式概念也已經過簡化，因此針對此項工具以外的相關程式設計軟體則不再本研究範圍內。此次實驗結果也不宜對其它程式設計軟體的教學成效做過度推論。

二、文獻探討

本研究主要探討不同的教學策略對基本程式概念學習成效的影響，以下分別就「兒童程式設計教學」、「認知負荷」、「完成問題教學策略」、「問題導向教學策略」四個主題進行資料搜尋及文獻探討。以期可以幫助研究者針對國小學生規劃出學習成效較佳的教學策略。

2.1 兒童程式設計教學

設計良好的教學策略以提升學生學習成效，是身為教育者常常會使用到的一種教學方法。在有關程式設計教學方面的研究，賴建二（2003）指出因為程式設計本質為一複雜的思考歷程，在教學的過程中涵括了許多概念，因此對於教學策略的研究相當分散，多數的研究僅針對某一部份進行探討，缺乏相關的整體研究。楊美菁（2005）更提到針對國小程式設計教學進行相關的實驗研究又屬少數，其中又以 Logo 程式語言最常為教育者使用。

以下針對幾位研究者以國小程式設計教學為主題來進行研究，將其研究結果概述如下：

許宏彰（2005）以國小學生為例，探討以 Logo 語言進程式設計時的心理活動過程，研究中透過問卷、晤談、原案分析等方式，瞭解學生使用 Logo 的思維歷程以及影響學生學習程式設計的相關因素為何。其研究結果發現，學生在使用 Logo 時有以下幾種典型的思維歷程，像是學生多採取順向思考的思維策略、Logo 的立即回饋功能有助於學生解題、學生對於模組化的觀念仍趨薄弱等。而在影響學生學習程式設計的相關因素方面，則是有以下幾點發現，像是指令的熟悉度、語言習慣、題意閱讀、程式模組化能力、發現規律能力、憶取舊經驗等。在研究最後也說明，學生對 Logo 程式語言的接受度提高，並指出在國小階段以 Logo 實程式語言教學是有其可能性的。

李昶龍（2006）針對國小六年級學生教導 MSWLogo 程式設計，並隨機將學生分為引導合作學習組、非引導合作學習組與個別學習組進行教學實驗，探討三組學生在學習成效的差異，在其實驗結果發現利用引導式合作學習有助於學生學習程式設計，而且隨著時間越長，教學內容越難，學習成效越顯著，而且此種教學方式對於低成就的學生在學習上的成效更為顯著。

楊書銘（2008）探討 Scratch 程式設計對六年級學生邏輯推理能力、問題解決能力及創造力的影響，針對國小六年級學生，修改 Polya 的四階段解題步驟，設計一套程式設計課程，透過「瞭解問題」、「分析問題」、「撰寫程式碼」、「檢視結果與除錯」四個步

驟進行教學。在其研究結果指出，此種的教學型態對學生整體的問題解決能力是有所提升，但詳細對問題解決能力的內涵作細分，除了「總分」與「猜測原因」這兩個項目高於控制組，但在「解釋推論」、「逆向原因猜測」、「決定解決方法」、「預防問題」這些分項則沒有明顯差異。

豐佳燕、陳明溥（2008）嘗試教導國小六年級學生使用 Stagecast Creator 來創作遊戲，此種程式語言為第一套融合了示範式程式設計（programming by example）概念，在其研究中透過同儕學習、遊戲創作與觀摩、評量等教學階段進行教學，其研究結果顯示有八成以上的學生認為 Stagecast Creator 中的範例程式有助於遊戲的設計，而有九成以上的學生認為程式設計不再是那麼令人畏懼的。

而林美娟（2005）在其國科會計畫—「以實驗策略增進初學者的程式設計概念與解題能力」的最後完整報告中指出，在其三階段的研究中，主要是針對程式設計教學的教材設計以及教學方法為主題，希望藉由實驗化教學軟體的使用，降低初學者的學習門檻。而第三年的研究著重在國小程式設計教學的實驗，主要目的是在找出適合國小學生學習的程式設計軟體，其中針對 Stagecast Creator、HANDS 及 Visual Basic 等三套軟體，進行教材設計，並針對國小四到六年級學生施行教學實驗。研究中利用「範例式教學」一步步讓學生瞭解基本的程式語法及該有的邏輯概念，並加上「遊戲式的軟體」引發學生的學習動機，也讓學生在有趣活潑的教學情境下進行學習。研究結果顯示，有近五成的學生覺得 Visual Basic 是他最喜歡的軟體，其次為 Stagecast Creator，三套軟體中以 HANDS 為最低。而在此次實驗中，有將近八成的家長對於國小加入程式設計的課程抱持樂觀的態度。

王麒富（2009）也以行動研究的方式，針對「應用直觀式 Scratch 軟體」設計教學課程，欲探討學生的問題解決能力是否有所提升。在其研究當中，給予學生問題解決的情境，依照問題導向的教學模式，讓學生經由「界定問題、分析問題、蒐集資料、提出解決方案、驗證與改進」的步驟進行教學。最後的研究結果學生的整體問題解決能力總分是有所提升，但經由其細分檢驗問題解決能力的五項指標來看，唯「問題覺察」、「原因推測」此兩分項成績達顯著水準，在「問題再定義」、「提出想法」和「尋求最佳方案」這三個向度並無明顯差異。研究者針對此種現象推究其因，發現當學生經過幾次失敗後，傾向直接尋求教師的協助，而不再自行尋找解決問題的方式。

Fain（2004）針對他的兒子 Dave（約 11 歲）進行 Java 的教學，他認為給予兒童愉快、有趣的學習環境，加上規劃良好的教學方法，兒童當然也能學習真正的程式設計，在此次實驗中 Fain 對兒童在學習程式設計歸納出有以下幾點特質：

- （1）多數的程式任務中需要最基本的「算術和代數技能」。
- （2）學生在四、五年級時已經發展了抽象推理能力。

(3) 兒童學的比成人快，但是他們並不具有「程式設計的相關經驗」，而這是一件好事，因為兒童不必從程序性的思考轉換到物件導向的思考。

(4) 成人可以持續的學習無聊的程式設計，但是兒童不行，每週 1~2 次的課程，每次 45 分鐘就足夠。

(5) 兒童在撰寫程式時，喜歡有立即的回饋；而相較於冗長繁複的程式，兒童較喜歡撰寫小段的程式碼。

(6) 圖形化的程式設計對兒童來說是最有趣的，甚至是大型的程式（像是：計算機、圈又過關遊戲或是桌球遊戲）都可藉此解釋給兒童聽。

而「學習」除了教師在教學面的策略設計外，另外針對合適的教學內容以及學生學習的軟體工具也是教學者需要注重的。楊美菁（2005）針對圖像式以及文字式的程式語言學習成效進行探討，指出兒童所學的程式語言應和專業的程式語言有所差異，要符合兒童的身心發展，並易於兒童操作使用，而研究結果在這兩種工具上學生學習成效並無顯著差異。在教學內容上，Hartmann 等人（2001）指出初學者在一開始學習程式設計，應該是在一個遊戲的環境，被教導最基本的程式概念，且不應該被複雜的程式語言介面所干擾。

綜上所述，教導兒童學習程式設計並非不可能，但在教學內容以及教學方法上必須有所限制以及事前規劃設計。因為程式設計本質上對兒童來說實屬複雜，因此教師在教學前所要考量的，教師必須針對學習者的特性，擬定良好的教學策略，並選擇適當的教學工具以及學習軟體，讓兒童在最小的干擾環境下，進行有意義的學習。

2.2 認知負荷

學習程式設計的過程中包含著問題解決技巧，而這是一種內隱的歷程，這些在學習上很難由學生自己覺察到，但卻是學習的關鍵（Govender & Grayson, 2006; Kirsner, 1998）。學者 Gomes 與 Mendes（2007）也指出，初學者學習程式設計最大的困難點不在於對程式語法的不理解，而在於不知如何結合自己所學來解決問題。

Fowler（1993）指出初階程式設計課程對於初學者最困難的是，初學者要同時處理以下這些事情，像是學習如何運用一般問題解決技巧、以演算法的思考規劃程式並學習運用程式設計的技巧、將心中所計畫的問題解決方案轉化為程式語言、熟悉程式語言軟體的介面環境。最後，學生還得確認程式是可以運作的，假使不行學生還要自己找出問題進行除錯。

鍾靜宜（2003）指出，在面對「問題解決」這樣的一個教學課題時，要考慮到的是問題解決是一個複雜的認知過程，在教學的過程中常常會造成學習者極高的認知負荷，

因此如果沒有良好的教學策略設計，學生在面對問題時常常會不知所措，容易迷失在問題中，因此導致不好的教學成效。

綜上所述，程式設計包含許多複雜的思考歷程，初學者除了要面對陌生的程式語法及指令、程式語言軟體介面，更為困難的是要將腦海所規劃的問題解決方法轉化為具體的程式指令撰寫，此種學習歷程常常會造成初學者的認知負荷（cognitive load）過高。

而以下以認知負荷的相關定義、理論進行探討，並歸納出相關的教學設計理論，以運用在之後的實驗教材設計。

Pass（1992）提出認知負荷是多向度的，他將認知負荷分為兩個向度：心智負荷（mental load）及心智努力（mental effort）。當個體意識到所要進行的工作是困難的或是要耗費很大的努力才能完成，就會對個體造成過多的認知負荷。

Sweller、Van Merriënboer 與 Paas（1998）在認知負荷與問題解決的研究中，指出傳統的問題解決法過於著重在解題的技巧，導致學習者需要耗費大量的認知能力在這方面上，沒有多餘的空間來進行基模（schema）的獲得與學習，因此造成學習者的認知負荷。而認知負荷理論最主要的目的在於基模的獲得與自動化，因此教師在進行教學設計時，應該將人類的認知結構加以考量（Van Gog, Paas, Jeroen J.G., & Van Merriënboer, 2008）。

在教學上，Sweller等人（1998）以教材設計的觀點，將認知負荷分為：（1）內在認知負荷（intrinsic cognitive Load）（2）外在認知負荷（extraneous cognitive load）及（3）增生認知負荷（germane cognitive load）三種，將其在教育上的意涵簡述如下（陳蜜桃，2003；Van Merriënboer, Kester, & Paas, 2006）：

- 1、內在認知負荷：即指教材內容的複雜程度，在教學上，如果學習內容過於複雜，以及學習內容間的元素，不是獨立的而是有相互交互影響的，因為學習者在其長期記憶內缺乏專業的知識（expertise），這樣的一個教材特性，會導致學習者在學習時工作記憶負載過多，導致內在認知負荷過高，影響其學習效果。
- 2、外在認知負荷：與教師如何組織教材以及呈現教材相關，不好的教學設計會導致外在認知負荷提高。
- 3、增生認知負荷：是幫助學習者建構基模用的，例如：教師設計一道未完成的數學題目，利用「引導學習」引導學生完成，此即為增加增生負荷（楊培渝，2006）。但教師所設法增加的增生負荷必須維持在認知負荷總量內，要適當的引入。

而教師在進行教學設計時，以認知負荷理論為基礎，就有不一樣的切入點（林宓雯，2007）。Sweller 等人在（1998）提出七大教學設計原則，主要是為了降低外在負荷，給

予教師在進行教材設計時參考使用。包括開放目標 (the goal-free)、示例 (worked example)、完全解題 (completion problem)、分心 (split-attention)、模態 (modality)、冗餘 (redundancy)、變化 (variability)。

而其中針對問題解決技巧的教學，Atkinson 和 Renkl (2003) 指出，利用範例教學會比利用問題解決來教導學生更為有效率。因此以下針對範例效應與完全解題教學進行說明：

一、範例效應 (worked example effect)

一般專家在解決問題時，因為具有足夠的基模 (先備知識) 可以供其使用，因此可以很容易全盤瞭解問題的狀態，而基模的自動化是指專家不用再費心思索問題解決的相關技巧，可以將全部的心力用於問題的解決上。而上述所說的這部分均是初學者所缺乏的，缺乏完整的先備知識，還要花心力在思索問題解決技巧上，導致認知負荷過重，而範例教學技巧，則是解決這樣困境的一個方法 (鍾宜靜，2003)。

範例教學主要是要來學習問題解決技巧的，通常在範例裡面會包含一個規畫好的問題，解題步驟，以及最後的答案 (Moreno,2006)。Atkinson、Derry、Renkl 與 Wortham (2000) 指出對於初學者來說一開始要獲得問題解決的技巧，利用範例閱讀的會比使用「問題解決」(problem solving) 的教學來的更有成效。由認知負荷理論來解釋即是藉由減少不需要的認知負荷，以較好的方式幫助基模建構及自動化 (Sweller et al.,1998; Van Merriënboer & Sweller, 2005)。而當學習者在初學程式設計這種有結構性 (well-structured domains) 的學習領域時，利用範例學習是非常關鍵的 (Atkinson & Renkl, 2003; Van Lehn, 1996)。

Garner (2003) 指出範例教學似乎是個理想教導初階程式課程的教學方法，就像我們並不會要求學生在沒看過英文短文前，就要求學生寫出英文短文。但這個方式卻有個缺點。要引起學生的動機，專注的坐在哪裡手動追蹤程式碼是有困難的。除非是有某些形式上的考試、測驗等，不然學生會想要略過閱讀的部分。但是如果方法激勵學生可以閱讀已經完成的程式碼，是可以幫助學生獲得之後問題解決要用的抽象的基模。而對於教師來說，創造出適合的範例程式，讓學生修改與擴充是個很耗時的過程。

二、完全解題效應 (completion problem effect)

所謂的完全解題技巧主要是將範例程式呈現一半，而學生要完成任務一定要透過閱讀教師給予的部分範例，才有辦法解題。此種技巧的好處在於強迫學生閱讀範例程式 (宋曜廷，2000)。

在程式設計的領域中，Van Merriënboer 和他的實驗夥伴 (Van Merriënboer, 1990;

Van Merriënboer & De Croock, 1992; Van Merriënboer, Krammer, & Maaswinkel, 1994; Van Merriënboer & Paas, 1990)做了相關方面的研究，認為在光是呈現範例讓學生觀察，學生在心智上還無法學習到抽象（mindful abstraction）的觀念，因為這種抽象的概念是需要學生將範例仔細的在心中運行過一遍，徹底理解後，將教師空缺的程式完成，這才能說學生是真正習得概念。

因此，這「半完成」（incomplete）的範例是非常重要的，必須具有良好的架構，並要有足夠的線索提供給學生，引導學生完成剩餘的程式。

綜上所述，程式設計的學習包含了問題解決的技巧在裡面，而對於初學者想要習得問題解決最有效的方法就是透過範例學習，但此次教學對象為國小學生，除了利用範例教學的技巧之外，為了克服學生在閱讀上的有可能產生的「不專心」，因此加入完全解題教學的技巧來引導學生學習，但其最終的目標仍是獨立完成一整個任務。

2.3 完成問題教學策略

完成問題策略是本研究中欲使用的教學策略之一，重點著重在一開始學習者針對完整範例（worked example）的觀察，以及第二階段學生完成教師給予的部分問題（completion example），最終的學習仍是希望初學者透過前兩步驟的引導可以獨立完成一個任務。以下針對此種概念的程式設計教學文獻進行比較探討，最後歸納整理出此次實驗中所使用到的「完成問題策略」。

2.3.1 相關文獻探討

Deimel 與 Moffat(1982)提出在指導初學者，應重視程式的閱讀，在當時提出” Ideal”的初階程式設計課程，將課程內容分為四步驟：(1) 給予學生可運作程式，學生觀察程式的動作，評判此段程式的優缺點。(2) 將具有良好結構（well-structured）的程式提供給學生，學生在此階段的主要活動是：閱讀並手動追蹤（tracing）程式。(3) 學生到了第三階段針對教師給予的程式修改並擴充，此階段的主要目的是藉由這些小規模的程式中，讓學生練習程式設計與編寫程式的技巧。(4) 最後的活動是學生自己完成一個程式，讓學生得以學以「應」用，練習之前所學到的程式設計及編寫程式的技巧。

Dalbey、Tournniare 與 Linn（1986）在當時認為在小學教導程式設計最大的困難點在於學生缺乏整體規劃的能力，這對學生在撰寫程式時，是很大的阻礙。因此，他們提出相似的教學策略，展示一段程式讓學生觀察，經由閱讀理解程式碼，可以幫助學生瞭解程式的整體運作，有助於在設計程式上的規劃。

Van Merriënboer (1992) 對於當時在初階程式設計課程的教學上，主要是教導初學者直接進行程式的編寫並產生新的程式。他認為這樣的方式不適合初學者學習，因此提出完成策略 (completion strategy)。其策略的重點擺在，初學者對於教師給予「結構良好的程式與運作中的程式」進行閱讀、修改和擴充。其研究結果發現，使用完成策略的學生有較佳的程式模版 (template) 產生，而在語法、語意的理解上，因為有較高程度的學習範例支持，也以完成策略組的成效較佳。

而張國恩及邱碧珠 (2000) 利用此種概念，設計初學者程式設計學習系統，研究者發現要初學者直接編寫程式，容易引起學習者的挫折感且失敗率極高。而在程式的撰寫上，每位學生的方式都不一樣，如果要讓此學習系統能針對不同的錯誤方式給予回饋是很困難的。因此，研究者採用「填空」(fill in) 的方式，使用範例搭配三種運算元：克漏填充運算元、修改運算元、擴充運算元，所謂的運算元指的即是不同的學習活動。在其研究結果發現，實驗組的學生在程式結構上有較好的表現，顯示所開發的初學者程式設計學習系統具有良好的示範作用。

Garner (2009) 同樣利用此種概念，設計教學平台 CORT，利用量化研究的相關數據支持 PCSM (Part-Complete Solution Method) 的教學方法，所謂的 PCSM 的教學方法指的是一種減低學生在學習程式設計時的認知負荷的一種方法。在其 CORT 教學平台中，課程的一開始，有一個完整的範例，範例中有解釋以及有一個問題在這個範例中，到了後面的課程，越來越多的部分是未完成的，學習者要根據任務解決例子中的問題。而在課程中間的過程中，會有不同完整程度的例子，而這些不完整的例子就像鷹架一樣幫助學習者學習。研究者以未學過程式設計的商學院學生為實驗對象學習 VB 程式語言，歷時 14 週，每週三小時。研究結果顯示，在學習成效上雖沒有顯著的幫助，但在學生學習的時間以及學習速度上是有提升的，而使用 CORT 的組別，在學習上所需要的協助也是比較少的，而這些對於學生的學習均是有幫助的。

綜上所述，利用範例學習搭配不同設計方式的「完成問題」，除了範例學習本身即有助於初學者在基模上的獲得之外；完成問題是將教師欲教導的概念，經過精心的設計融入其中，對於學生抽象概念上的學習，更能幫助學生理解。因此，針對研究對象為國小四年級學生，考慮其先備知識不足以及上述初學者所會有的學習困難，研擬一套完成問題教學策略，而非多數常使用的問題導向教學，利用此種方式，將抽象的概念教予學生，期能達到較佳的學習成效。

2.3.2 完成問題教材設計

範例教學在一開始的確可以降低認知負荷，提高學習成效，但是過多過久會造成無效的負荷、並阻礙學習者的學習進展。Kalyuga、Ayres、Chandler 與 Sweller (2003) 指出當學習者獲得更多的專家知識後，會出現一種現象叫做專家反向效應 (the expertise reversal effect)。舉例來說，學生在專家知識不足的情況底下，透過範例來學習會比利用問題解決學習還有成效，但相反的，如果學生已經具備足夠的專家知識，就會造成反效果。這樣的原則對應到問題解決相關的教學上，給予教師在設計教學策略時，應該在一開始利用範例教學導入，但隨著學生專家知識護得的越多，應該逐步減少範例的分量，慢慢讓學生面對獨立問題解決的學習情境。

所以在研究中的教學策略設計為，搭配範例進行教學並運用「步驟逐漸向後減少」(backwards fading) 的技巧，隨著課程的進行，逐步減少範例中的元素，最後學生會獨立完成一個完整問題 (Clark et al., 2005)。而所謂的步驟逐漸向後減少技巧，即是當「完成問題」逐漸形成「獨立任務」的過程中，學習者會一步一步完成被省略的地方，而且學生完成的步驟的數量是越來越增加的 (Renkl, Atkinson, Maier, & Staley, 2002)。

2.4 問題導向教學策略

以問題解決的步驟為基礎，提供學生必要的線索與指引，並餘留空間讓學生得以探索、思考，而在這過程中學生學到問題解決的技巧。(鐘靜宜，2003)。而問題解決並非程式設計領域獨有的，因此有研究者以通用的問題解決歷程歸納出適合於程式設計領域的問題解決步驟。

陳明溥 (2007) 在其實驗中，以「擲骰子遊戲」為其程式設計的任務，並歸納程式設計領域的問題解決步驟為其教學方法，步驟分為四部份：定義問題、規劃解題方案、進程式撰寫、及驗證除錯。楊書銘 (2008) 將以 Polya 的四階段解題法，應用於其研究中，設計一套程式設計的課程，研究者將解題模式稍加更改為：瞭解問題、分析問題、撰寫程式、檢視結果與除錯。

而在本研究中，問題導向策略為所欲使用的策略之一，因此依照之前的研究者針對程式設計領域所歸納的問題解決步驟，並針對考量國小四年級學生的能力，更改為以下四步驟：瞭解問題、分析問題、撰寫程式區塊、運作與觀察程式。

三、完成問題策略教材設計理論模型

3.1 模型說明

本研究中的完成問題策略，主要根據以下理論模型（圖 1）進行教材上的設計，課程發展順序依照模型中的金字塔底部往上走，依序為追蹤觀察、模仿修改以及擴充創作，各階段說明如下：

（1）追蹤觀察

Atkinson 等人（2000）指出對學習者來說一開始要獲得問題解決的技巧，利用範例的閱讀會比使用問題解決的教學來的更有成效。而在追蹤觀察活動中，教師設計適當的範例，透過範例的引導，學生藉此習得程式概念的相關基模。此舉對應到認知負荷理論，即是減少會干擾學生學習的外在認知負荷，透過範例的觀察增加學生的增生負荷，用以幫助學生基模的建構與自動化（Van Merriënboer, et al., 2005）。

（2）模仿修改

Kalyuga 等人（2003）指出當學習者獲得較多的專家知識後，則會出現專家反向效應，此時再給予學生過多的範例，反而會形成反效果。而在一開始的追蹤觀察活動，學習者透過觀察範例習得程式概念，但第二步驟應將範例內容逐步減少，而有意義的範例學習，也必須仰賴學習者願意花心思、專注的閱讀範例（Siti Soraya Abdul Rahman & Benedict du Boulay, 2010）。

因此，在教學的第二步驟模仿修改，依據 Clark 等人（2005）所揭示的步驟逐漸向後減少（backwards fading）技巧，隨著課程的進行將範例所包含的程式逐步減少，並考量到實驗對象為國小四年級學生，將此階段的範例轉換為「部分完成的範例」，學生必須藉由閱讀教師給的部分程式，透過模仿或是修改的動作，才能將此階段的學習任務完成。

（3）擴充創作

而在程式教學上，最終目標仍期望學生能透過程式設計來獨立解決問題（Winslow, 1996）。因此，在第三步驟擴充創作上，搭配遊戲設計，教師給予學生一個獨立的問題，此問題與遊戲能否順利完成息息相關，主要目的是希望學生能透過上兩步驟所習得的概念，實際應用在問題解決的情境中。但仍會針對實驗對象的特性加以考量，提供學生基本會使用到的程式指令，讓學生加以組合使用，並鼓勵

學生勇於創作思考。

綜合上述，將理論及教學策略對應到實際的教學活動上。教師於課堂中給予學生的任務會隨著課程的進行由範例(worked example)演變為完成問題(completion example)，最後學生會被要求獨立解決一個問題(full problem)。對應至具體的教學活動即為追蹤觀察、模仿修改與擴充創作，這三個活動正為完成問題教學策略的核心內涵。而學生在這三個教學活動中的角色定位演變，從觀察者到模仿者，最終角色則為能獨立完成問題的完成者。此最後學習活動也正好與學生所應具備的十大基本能力相符合——培養獨立解決問題的能力。

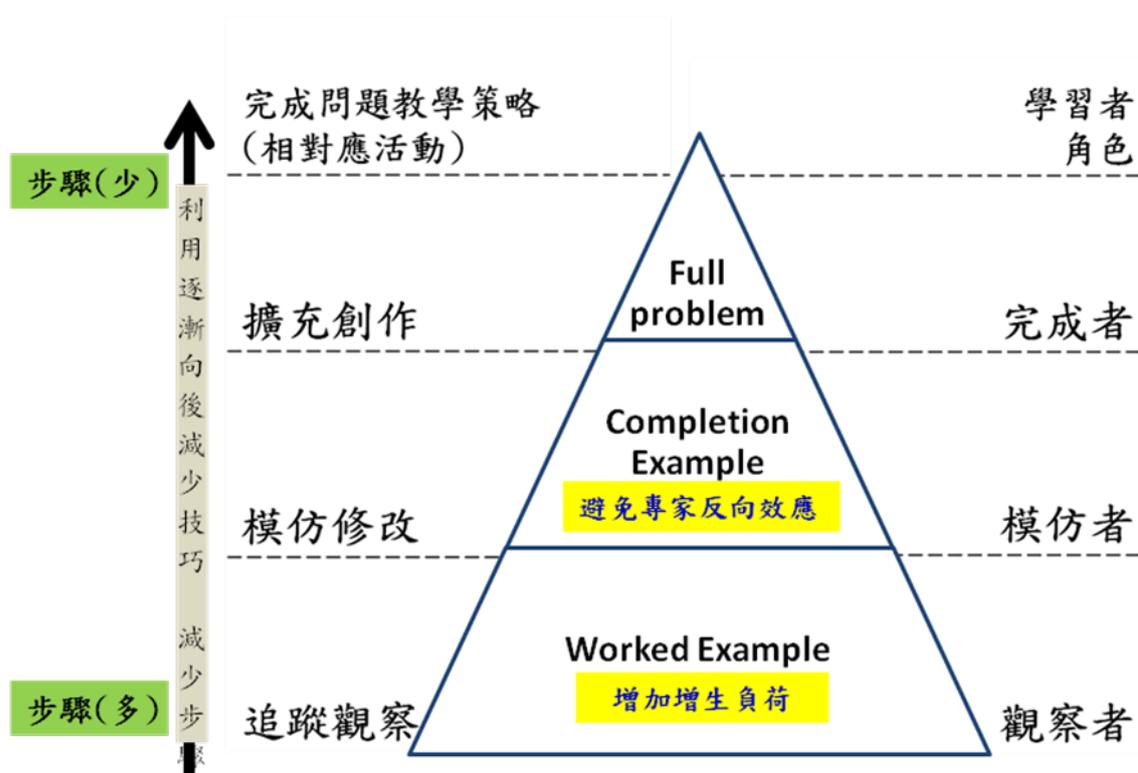


圖 1 完成問題策略教材設計理論模型

資料來源：參考自 Clark,et al., (2005:199)。

3.2 實際應用

此小節說明上述模型應用在實際教材的情形，此次實驗對象為國小四年級學生，為了讓教材可以符合學習者的特質，因此在教學活動上做了以下設計：

3.2.1 追蹤觀察

(1) 設計理念

在此活動中主要的意涵是希望學習者可以藉由觀察、追蹤範例程式，給予學生專家的模型，增加其增生負荷，在張國恩及邱碧珠（2000）中設計的初學者程式設計學習系統中，則是利用克漏填充運算元的策略，將範例程式中的主要概念空下，讓學生填空，藉此強迫學生勢必要觀察完整個範例才能進程式碼的撰寫填空。但考慮到此次教學對象為國小學生，因此將學生在此活動的角色定義為「觀察者」，經由自行手動播放範例程式，觀察範例程式的運作，透過學習單的輔助，選出學習單上哪個程式區塊最接近範例程式，藉此讓學生可以觀察程式的運作，習得基本程式概念，並可將所得到的概念與實際程式區塊相結合，轉化為具體的程式指令。以下利用「順序平行單元」為例，說明追蹤觀察技巧在實際教材的應用。

(2) 範例說明

圖 2 中四張圖為範例程式的動態截圖。實際上，學生會經由執行範例程式看到角色的連續動作。程式一開始小恐龍從舞台左下角移動到舞台中央，如圖 2 (A)到圖 2 (B)，接著小恐龍會說話，如圖 2 (B)，最後小恐龍在原地改變大小，由大變到小，如圖 2 (C)、圖 2 (D)所示。

範例程式運作方向：學生透過觀察範例程式的運作，可得到順序執行的概念

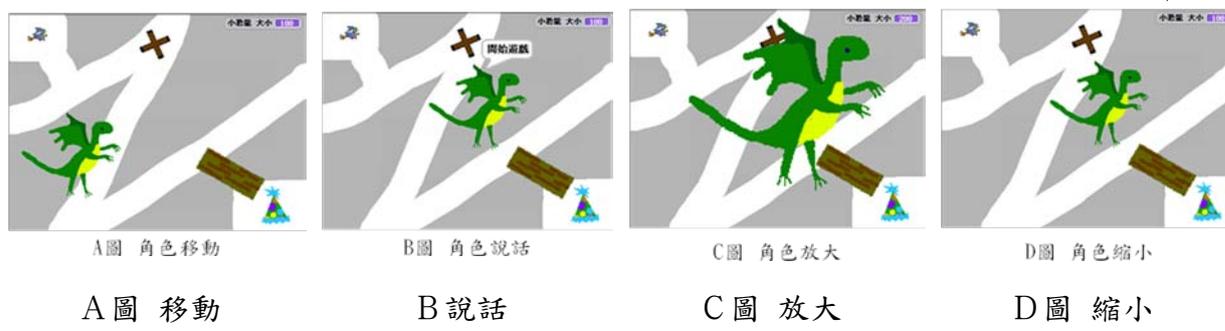


圖 2 追蹤觀察範例說明

這一連串的动作主要透過視覺化的設計，讓學生實際感順序執行概念，並搭配學習單上問題（如圖 3），找出角色动作的順序性，藉此傳達指令的安排是要有組織有順序的。

小恐龍的動作有：移動、放大、說話		
哪一張圖最接近小恐龍的指令，把你覺得對的答案，在 <input type="checkbox"/> 內打勾，答案只有一個		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

圖 3 追蹤觀察學習單範例

3.2.2 模仿修改

(1) 設計理念

在此部分的教材主體是「部分完成的程式」，讓學生針對教師留下的空白加以修改、完成。但是針對國小學生，在教材設計上，會接續上一程式，讓學生有所對照，避免造成過多的認知負荷，因此會利用活動一的程式為基礎，給予學生含有部分程式的練習題。讓學生模仿所觀察到的程式（圖 2）及觀察教師給予的部分程式（圖 4），完成練習題中的任務。

(2) 範例說明

如圖 4 所示，學生點開 Scratch 程式即可看到任務以註解說明的方式附在裡面，下方為活動一的程式區塊，此例是以程式的修改為學生主要活動。在活動一中小恐龍的程式執行順序從移動、說話到最後在原地放大縮小。因此，在模仿修改活動中，要求學生更動程式執行的順序，將小恐龍的順序改為先變大變小，再移動到別處，接著說話（詳見圖 4 黃色任務說明）。且因考慮到學習者為國小學童，因此在舞台中央附上小恐龍欲移動的座標點，以減少學生在完成此段程式所會遇到的障礙。

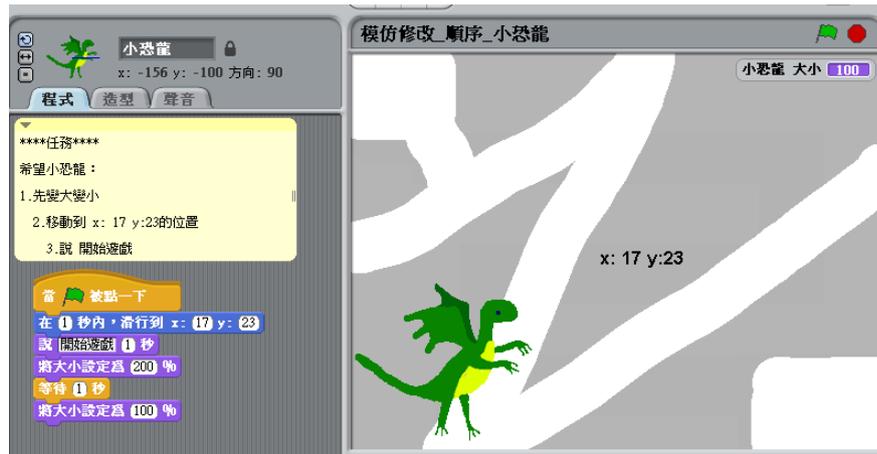


圖 4 模仿修改活動

3.2.3 擴充創作

(1) 設計理念

在活動三中，直接讓學生進行電流急急棒遊戲任務的完成。在任務題當中，提供學生關鍵的指令，如 repeat、if-then 等，以及相關指令，讓學生利用自己所學到控制結構的概念，並自由的加入依照自己的想法，來完成任務（如圖 5 所示）。在此活動中，學生可以選擇是否要使用教師給予的指令。

(2) 範例說明

此次實驗的最終應用為學生能習得基本程式概念（控制結構），並加以運用在遊戲設計中，因此在此例中，學生主要能創作一隻會說遊戲規則的小恐龍，所使用的程式概念為順序執行。學生必須讓小恐龍先顯示，接著說明遊戲規則，最後小恐龍要消失。在此時學生所要面臨到的就是一個待解決的問題情境，在任務說明下，則附上基本的指令，而教師在實際教學時，則會鼓勵學生創作發會自己的想法，加入想要的指令，增加角色的趣味性，但任務當中所賦予小恐龍遊戲上的功能，學生也必須加以完成。



圖 5 擴充創作活動

四、研究方法

本研究旨在探討不同的教學策略對學生學習基本程式概念學習成效的影響，教學內涵為「基本程式概念—控制結構」，以下就「研究流程」、「研究設計」、「研究對象」、「研究工具」、「實驗程序」及「資料分析」進行說明。

4.1 研究流程

為了清楚明瞭研究流程，研究者將其區分為三個階段：準備階段、教學實驗階段以及完成階段，分別描述於下。進行流程如圖 6 所示：

4.1.1 準備階段

(1) 資料搜尋

廣泛蒐尋程式設計教學與相關教學策略的文獻資料，同時與指導教授與多位國小教師經驗交流之後，確定此研究方向，並依此設定研究目的與待答問題。

(2) 擬定研究主題

主題確定後開始進行文獻探討主要的內容為程式設計教學、兒童程式設計、初學者的學習困難、程式設計教學策略、基本程式概念以及各類學習理論、認知負荷理論。依照相關文獻以及與指導教授和多位國小教師討論後，開始著手進行教學策略的擬定及教案的撰寫。

(3) 設計相關教學策略及教案撰寫

擬訂教學內容為基本的程式概念（即簡單的控制結構：包括順序平行、重複執行、條件執行）。而教學策略部分，除了涵括目前在教導兒童程式設計常使用的問題導向教學策略以及一般教師進行教學上會使用到的「指令講述—程式撰寫」的教學方式。而此次使用的第三種策略—完成問題教學策略，是參考文獻、教授、專家討論過後，改良多位學者使用過的完成策略訂定而來。

(4) 發展教學相關教材

根據策略的內容及相關策略活動主軸，分別設計不同策略的教材與教學媒體。

4.1.2 實驗階段

研究採取準實驗設計的方式進行，此為整個研究實驗活動進行的主要階段。由研究者為實際教學教師，在實驗班級中進行教學實驗。本研究中所有研究數據資料之取得，皆在此階段內完成。

4.1.3 資料分析階段

此階段以 SPSS 為主要統計工具，彙整學生的相關測驗成績與學習態度問卷進行量化分析，研究者依照研究目的選用適當的統計方法加以分析，最後根據分析結果提出結論與建議，撰寫研究報告。

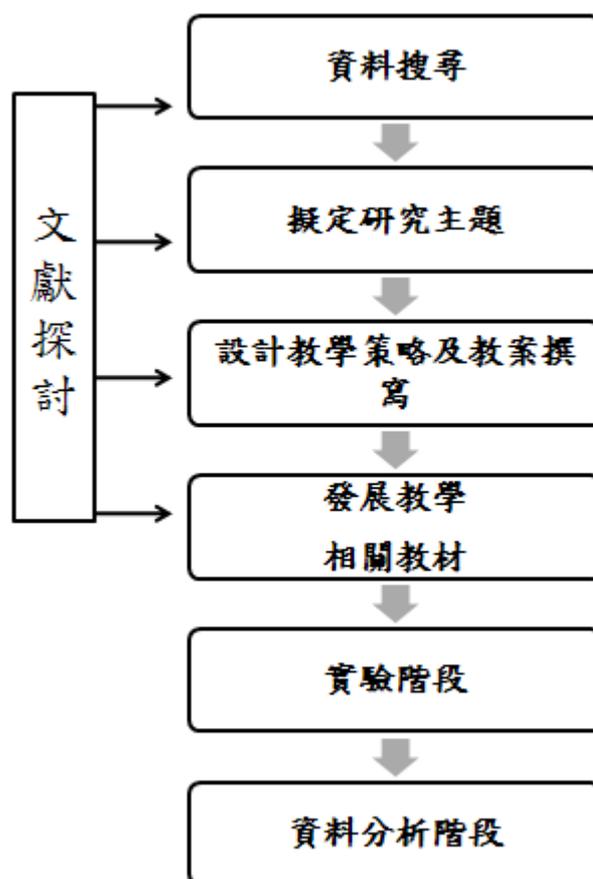


圖 6 研究流程圖

4.2 研究設計

本研究採準實驗研究法，藉由準實驗設計，以瞭解不同的教學策略對國小四年級學生學習基本程式概念學習成效的影響。實驗架構如圖 7 所示：

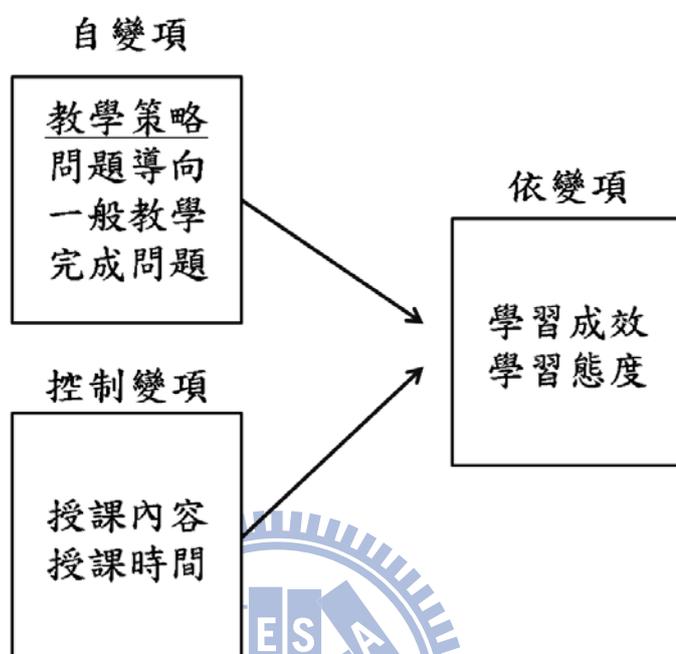


圖 7 實驗設計架構圖

4.2.1 研究變項

一、自變項

以「教學策略」為自變項，其中包含三種策略的使用，分別為「問題導向教學策略」、「一般程式教學」、「完成問題教學策略」。

「問題導向教學策略」將教學活動分為「瞭解問題」、「分析問題」、「撰寫程式」、「觀察與運作程式」四部分，以問題導向的方式引導學生解決問題並完成任務。「一般程式教學策略」主要教學活動是以教師講解指令用法，學生在教師指令說明之後直接進行任務的完成。「完成問題策略」的教學活動主要分為「追蹤觀察」、「模仿修改」、「擴充創作」三部分，在最後「擴充創作」的部分，依照教師給予的指令並自行創作完成任務。三種策略均以完成「電流急急棒」遊戲為學習主軸。

二、依變項

研究中的依變項有兩個「學習成效」、「學生學習態度」，其中「學習成效」包含各單元的形成性測驗與總結性測驗，其中各單元形成性測驗內容包括：(1) 紙筆測驗：該單元主要程式概念試題分數 (2) 程式實作：該單元任務評分。

總結性測驗包括：(1) 紙筆測驗：包含「順序平行」、「重複執行」、「條件執行」三項的基本程式概念綜合試題 (2) 程式實作：電流急急棒遊戲評分。

而學生學習態度包括：(1) 學生情意態度：其中細分為有趣性、愉悅性 (2) 學習動機 (3) 各單元難易度：總共三個單元 (4) 不同策略在學生學習上的幫助程度。

三、控制變項

為了減低自變項以外的其他變項對實驗效果產生影響，本研究在實驗過程中盡量掌控除了自變項之外的其餘教學條件，並使其盡可能相同。實驗進行時三組之教學者皆由研究者親自擔任，三組教學時間同為 11 節課，共 440 分鐘、教學內容均以基本程式概念為主，而三組學生實際感受到學習目標的同為設計「電流急急棒遊戲」，每次教學進度步驟均保持同步，實驗者所操縱的僅為三組教學策略上的不同。



4.3 研究對象

本研究以臺北市某國小四年級學生作為研究樣本，參與學生共三個班、91 位同學，為配合原班級之課程安排，以上課班級為單位，分派為問題導向組、一般教學組以及完成問題組三組進行教學實驗。此次教學實驗分組人數分配如表 1 所示：

表 1 實驗各組人數分配表

教學策略	分組人數
問題導向組	32 (男生 16 人; 女 16 人)
一般教學組	31 (男生 17 人; 女 14 人)
完成問題組	29 (男生 15 人; 女 14 人)

每位同學在三年級均學習過基本電腦使用(如：鍵盤滑鼠操作、基本視窗使用、網際網路的使用)，並於四年級上學期學習 Word 文書處理軟體。而對於此次教學的主軸「基本程式概念」均未曾學習過。

在教學實驗前，利用學生「上學期電腦課平均成績」用以檢驗學生在電腦基本操作能力上是否同質。使用 SPSS 軟體進行變異數分析，先以 Levene 法進行變異同質性檢定，檢定值為 1.003， $p>.05$ ，未達顯著水準，符合變異數同質性假設。而透過變異數分析， $F=.054$ ， $p>.05$ ，未達顯著水準。

表 2 各組上學期電腦科平均成績變異數分析

	平方和	自由度	平均平方和	F 檢定	顯著性
組間	.593	2	.296	.054	.948
組內	486.344	88	5.527		
總和	486.937	90			

* $p<.05$

由表 2 得知，三個班級的學生在「上學期電腦科平均成績」上並無顯著差異，推估三班在電腦基本操作能力上無顯著差異。

4.4 研究工具

本研究使用的研究工具包括 4.4.1 教學實驗相關教材、4.4.2 基本程式概念測驗、4.4.3 學習者態度問卷，各工具內涵詳述如下：

4.4.1 教學實驗相關教材

此次教學實驗的三組的學習內容均為控制結構，而學生最終的學習任務也以完成「電流急急棒」遊戲為目的，以下先就教材分析與 Scratch 程式設計軟體分述如下。再根據不同教學策略的教材內容進行說明。

一、教材分析

表 3 為此次教材分析一覽表，詳細說明如下：

表 3 教學單元與教材分析表

教學主軸：基本程式概念—控制結構		
教學單元	相關程式概念	具體遊戲任務
順序平行	<ul style="list-style-type: none">● 依序執行程式 (sequence)● 撰寫兩種獨立且並行的事件 (parallel)	創作會說規則的小恐龍
重複執行	<ul style="list-style-type: none">● 計次型的重複 (repeat)● 無限型的重複 (forever)	設計遊戲障礙物
條件執行	<ul style="list-style-type: none">● 利用條件敘述 (if 及 if-else)，設定該有的執行條件	訂定遊戲規則

現階段國小的資訊課並無相關的程式設計課程，因此針對教學內容上，以研究者自編教材為主，教導的內容以程式基本的控制結構為主要教材設計的主軸。

搭配所使用的程式語言軟體—Scratch，以下細分為三個子單元，第一單元為順序平行單元主要學習目的是希望學生可以用系統性的思考方式將想要執行的步驟依序執行；且會運行兩個堆疊區塊 (stacks)，創作兩種獨立且並行的事件。第二單元為重複執行，簡單的分為計次型的重複 (repeat) 以及無限型的重複 (forever)，主要學習目的希望學生利用重複執行來重複運作一系列的指令並精簡自己的程式。第三單元為條件執行主要學習目的是希望學生利用條件敘述 (if 及 if-else)，將程式的處理流程依照實際遊戲運作的狀況，設定該有的執行條件。

在教學設計上，搭配設計遊戲，讓學生運用學習到的控制結構逐步將遊戲完成。相關遊戲概念如圖 8 所示：

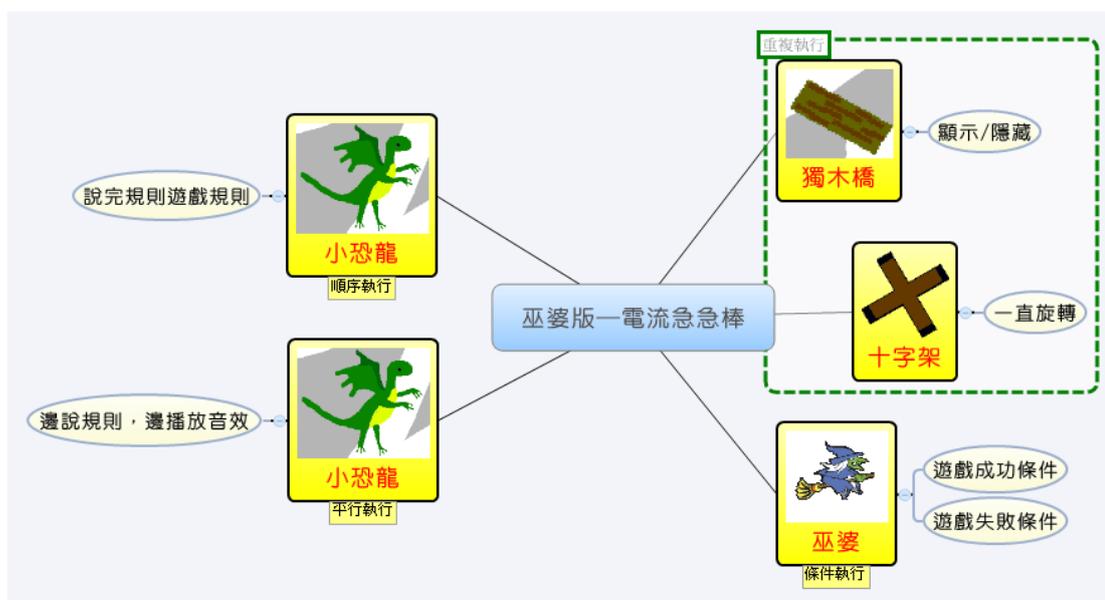


圖 8 電流急急棒遊戲與控制結構程式概念結合圖

依據上述教學內容以及設計「電流急急棒遊戲」為此次學習主軸，針對不同的教學策略，進行教材設計。

二、Scratch 程式設計軟體

Scratch 為美國麻省理工學院 (MIT) 的 Lifelong Kindergarten group 媒體實驗室所開發的一套視覺化程式設計軟體。Scratch 採用拖曳積木、堆疊積木的概念，免除打指令的步驟，讓兒童視覺化的拖拉程式區塊、排列出想要的程式，如：動畫、遊戲、故事等。在本研究中即採用此套程式設計軟體，除了其介面親切容易入門，加上此次實驗使用的是 2009 年發佈的 Scratch 1.4 版，此版本對於中文的支援度更大，因此可避免學生對於英文指令的障礙。

在其官方 1.4 版參考手冊中開門見山即說明 Scratch 是一個新的程式語言，此套軟體雖然是為兒童設計使用，但基本的程式概念卻相當齊全，舉凡像是順序 (sequence)、迴圈 (looping)、條件敘述 (conditional statements)、變數 (variables)、並行執行 (parallel execution)、布林邏輯 (boolean logic)、亂數 (random numbers)、事件控制 (event handling) 等，均包含在內。

在研究中欲教導學生「基本程式概念—控制結構」，此軟體既包含了基本的程式概念，又可藉由簡單的介面使用來導入核心概念避免學生被繁雜的語法學習所牽絆，因此採用之。



圖 9 Scratch 1.4 版 界面說明

Scratch 還包含了對於多媒體素材的支援，學生除了可以利用內建的繪圖工具繪製自己喜歡的角色，針對圖片匯入 Scratch 可容納 JPG, BMP, PNG, GIF 等的格式。在聲音上學生還可以錄製自己的聲音，或是匯入聲音檔（MP3、WAV 等），在這次電流急急棒的遊戲設計主題中，此項特點也增加兒童在學習程式設計的樂趣。

以下將各教學單元的程式概念與 Scratch 中的相關指令整理如表 4：

表 4 教學單元程式概念與 Scratch 指令範例表（續）

教學單元	Scratch 指令範例	
單元一順序平行 分為順序 sequence 與平行 parallel	 <p style="text-align: center;">圖 A</p> <p style="text-align: center;">單純順序執行</p>	 <p style="text-align: center;">圖 B</p> <p style="text-align: center;">圖 A、B 同時執行為平行執行</p>

<p>單元二重複執行</p> <p>分為計數型 repeat 與無限型 forever</p>	 <p>計數型 repeat</p>	 <p>無限型 forever</p>
<p>單元三條件執行</p> <p>分為 if 與 if-then</p>	 <p>如果 if</p>	 <p>如果...否則 if-then</p>

三、各策略教材設計

本研究的實驗教學由研究者擔任施測者與教學者，而此次實驗對象為國小四年級學生，在教學實驗前並無相關程式設計課程的學習經驗，因此在課程的前三週教導學生熟悉 Scratch 程式設計軟體，以及相關基本指令的應用。

在第四週以後，學生接觸到的教材內涵是一樣的，均以設計電流急急棒為主要學習目標，研究者實際所操控的僅為教學策略上的不同。以下分別就一般教學組、完成問題策略與問題導向策略的教學活動與教材設計進行說明。

(一) 一般程式教學組

在一般程式教學中，主要教學活動先講解基本程式概念，接著教師針對關鍵的指令進行說明，如重複執行單元中的 repeat 與 forever、條件執行單元中的 if 與 if-then。說明完後學生進行基本的指令練習，並於第二堂課，直接進行電流急急棒遊戲程式的撰寫。每單元教學時間為兩堂課，相關教學流程圖如圖 10 所示。

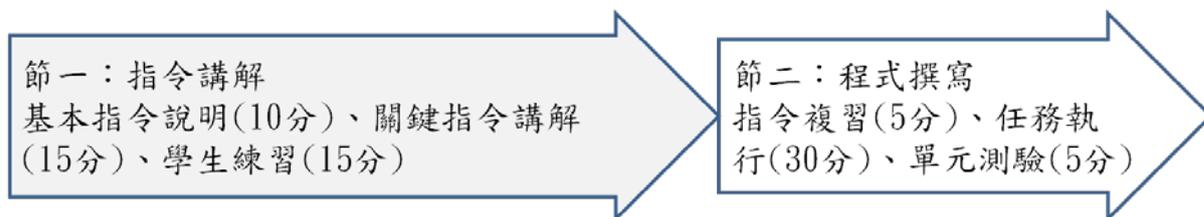


圖 10 一般程式組教學流程圖（兩堂課）

(二) 完成問題策略組

在完成問題教學中，以策略的三個子活動為主軸，分別設計相關的範例程式、學習單、完成程式、以及搭配遊戲設計進行擴充創作的任務。每單元教學時間為兩堂課，以圖 11 說明策略運作的教學流程：

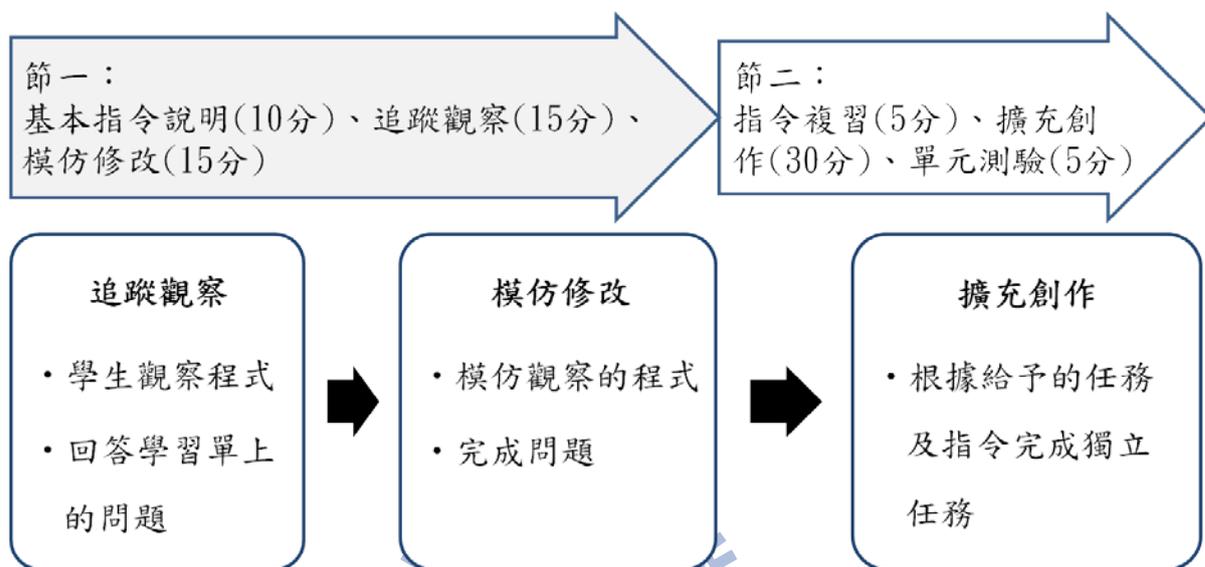


圖 11 完成問題組教學流程圖 (兩堂課)

1、課堂活動說明

(a) 追蹤觀察活動：在課程一開始，讓學生經由觀察已經設計好的運作程式，藉由觀察程式的運作過程配合上學習單，將範例程式與範例程式的原始程式區塊做結合呈現給學生，學生在學習單上勾選出正確的答案。學生在此階段以範例 (worked examples) 觀察為主要的任務型態。

(b) 模仿修改活動：第二階段，以「追蹤觀察活動」的程式為基礎，給予學生含有部分程式的練習題，學生在此階段以完成問題 (completion example) 為主要的任務型態。學生實際動手拖曳程式區塊，透過模仿範例程式以及觀察教師給予的部分程式區塊，來完成練習題中的任務。

(c) 擴充創作活動：當學生習得基本程式概念之後，再給予過多的範例或是提示會產生專家反向效應，但考量到學生為四年級學生，因此在此階段以完成獨立問題為主要任務型態，但在練習題中會給予學生部分基本指令，此部分基本指令並未組合而是散亂排列的讓學生挑選自行排列組合，用意提醒學生有此種指令可使用。教師在此部分的活動會鼓勵學生除了使用給予的指令外，也期望可以自行發揮創意加入新的可用指令完成任務。

2、完成問題組教材範例

此次教學內容共分為三個子單元順序平行、重複執行以及條件執行，以下以條件執行單元的實驗教材為範例，如表 5、表 6、表 7 所示，其他單元詳見附錄一。

表 5 追蹤觀察活動—條件執行單元為例（續）

<p>開始畫面：播放順序由圖 A 到圖 B，以下說明範例使用的方法。</p>		
 <p>圖 A</p>	 <p>圖 B</p>	
<p>教材分析</p> <p>條件執行 (if)</p> <p>說明： 當學生點角色（巫婆），角色會移動到滑鼠上，學生必須像玩電流急急棒遊戲一樣，點主角巫婆，碰觸其它角色，觀察角色的動作。</p> <p>範例程式說明： 圖 C 為角色（巫婆）碰到角色（十字架）的情況：碰到角色（十字架）角色（巫婆）說：扣 1 分。</p>	<p>教材內容</p>  <p>圖 C</p> <p>範 例 結 束</p>	
<p>學習單內容：巫婆 碰到十字架 就 說扣 1 分。根據說明以及觀察到的情形，選出正確的程式圖。</p> <p style="text-align: center;">問題二：如果....就.....</p> <p style="text-align: center;">下面哪一張圖，最接近角色(巫婆)的程式，在□內打勾 狀況：巫婆 碰到十字架 就 說扣 1 分</p>		
 <p><input type="checkbox"/></p>	 <p><input type="checkbox"/></p>	 <p><input type="checkbox"/></p>

教材分析	教材內容	
<p>條件執行 (if-then)</p> <p>範例程式說明：</p> <p>碰到角色 (紅妖怪) 就放大</p> <p>沒有碰到角色 (紅妖怪) 回復正常</p>	<p>圖 A 角色 (巫婆) 未碰到角色 (紅妖怪)</p>	
	<p>圖 B 角色 (巫婆) 碰到角色 (紅 妖怪)</p>	
	範 例 結 束	
<p>學習單內容：巫婆碰到紅妖怪 就 變大 否則 變回原狀，根據說明以及觀察到的情形，選出正確的程式圖。</p>		
<p>任務(一)動動眼：點開桌面上的「巫婆奇遇記」，拿起角色(巫婆)碰碰看其他角色</p> <p>問題一：如果... 就.... 否則...</p>		
<p>下面哪一張圖，最接近角色(巫婆)的程式，在□內打勾</p> <p>狀況：巫婆碰到紅妖怪 就 變大 否則 變回原狀</p>		
<div style="border: 1px solid black; padding: 5px;"> <p>當 巫婆 被點一下</p> <p>重複執行</p> <p>如果 碰到 紅妖怪</p> <p>將大小設定為 100 %</p> <p>否則</p> <p>將大小設定為 40 %</p> <p><input type="checkbox"/></p> </div>	<div style="border: 1px solid black; padding: 5px;"> <p>當 巫婆 被點一下</p> <p>重複執行</p> <p>如果 碰到 紅妖怪</p> <p>將大小設定為 40 %</p> <p>否則</p> <p>將大小設定為 100 %</p> <p><input type="checkbox"/></p> </div>	<div style="border: 1px solid black; padding: 5px;"> <p>當 巫婆 被點一下</p> <p>重複執行</p> <p>如果 碰到 小鬼</p> <p>將大小設定為 100 %</p> <p>否則</p> <p>將大小設定為 40 %</p> <p><input type="checkbox"/></p> </div>

表 6 模仿修改活動—條件執行單元為例

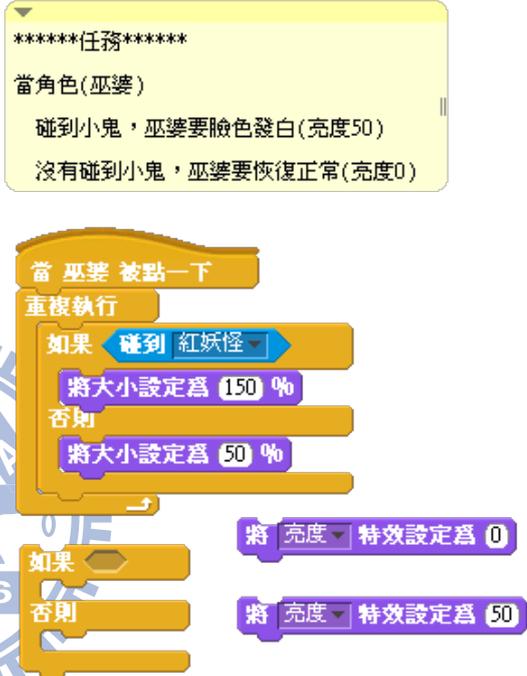
	條件執行 (if)	條件執行 (if-then)
教材分析	<p>完成問題內涵： 給予角色動作，學生根據任務加入偵測的條件。</p> <p>學生任務： 當角色（巫婆）碰到帽子，巫婆要說：過關了。</p>	<p>完成問題內涵： 給予角色動作以及 if 基本框架，學生根據任務加入偵測的條件。</p> <p>學生任務： 當角色（巫婆）碰到小鬼，巫婆要臉色發白，沒有碰到小鬼，巫婆要恢復正常。</p>
教材內容	 <p>上圖為條件任務的舞台圖</p> 	

表 7 擴充創作活動—條件執行單元為例

教材分析	教材內容
<p>條件執行 (if & if-then)</p> <p>擴充創作任務： 給予學生巫婆.sprite 鼓勵學生自己更換角色造型，利用條件執行，創造自己的遊戲規則。</p> <p>教師給予的指令： 全部停止（遊戲過關/失敗的條件） [偵測]當碰到顏色■ （利用顏色來設定遊戲規則）</p>	

(三) 問題導向教學組

問題導向組的學習教材以設計電流急急棒遊戲為最終目的，教學流程即是問題解決的步驟，分別為：瞭解問題、分析問題、撰寫程式、運作與觀察程式，而每單元教學時間為兩堂課，如圖 12 所示：

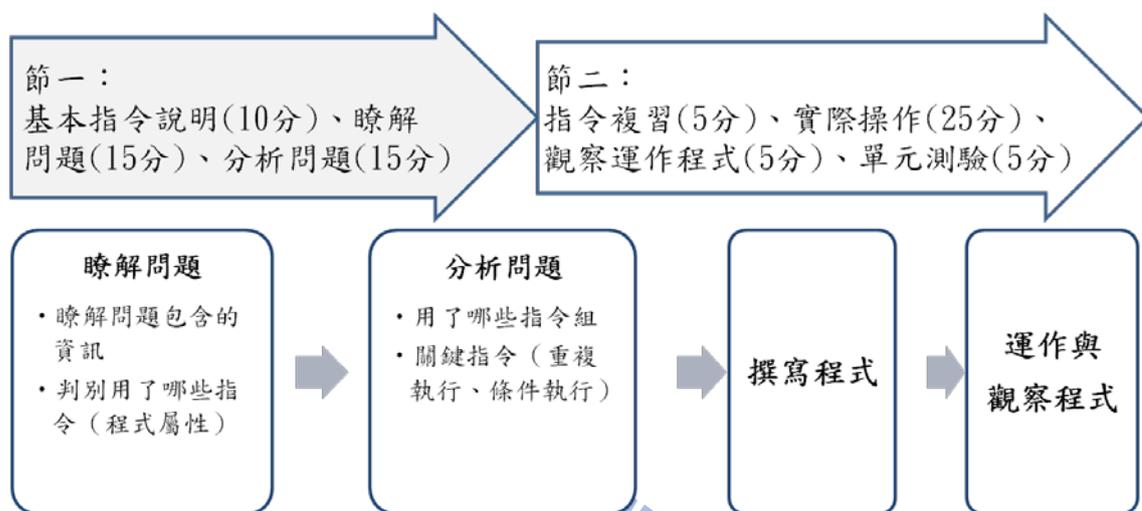


圖 12 問題導向組教學流程圖 (兩堂課)

教學內容共分為三個子單元順序平行、重複執行以及條件執行，以下以順序平行單元為範例進行說明，詳細教學內容見附錄二：

1、瞭解問題：觀看遊戲中的第一個任務「會說規則的小恐龍」，詢問學生看到的小恐龍有哪些動作？分別使用到哪種屬性的指令？(如：外觀、移動等)給予學生任務：依照看到的例子，排列出你的小恐龍。

2、分析問題：透過全班討論教師帶領學生分析問題。主要問題環繞著程式中使用哪些指令，以及關鍵指令的導入，透過問題的引導撰寫程式。以下為教師會詢問學生的相關問題：

- (1). 請問這些步驟利用到哪些「程式盒子」？
- (2). 分別是哪些指令？
- (3). 會播放聲音嗎？
- (4). 會放大縮小嗎？
- (5). 分別是哪些指令？(說話、顯示隱藏)
- (6). 如果小恐龍要一開始出現，之後消失，那程式要怎麼排列。(關鍵概念：順序執行)

3、撰寫程式：教師準備好練習檔案學生開啟後，進行實際程式撰寫。

4、運作與觀察程式

4.4.2 基本程式概念成就測驗

Oliver (1993) 將程式設計知識分為兩大主要類別，一為概念性知識 (conceptual knowledge)，一為策略性知識 (strategic knowledge)。概念性知識是指瞭解程式的構造、語法；而策略性知識意味著剖析問題與詳述問題的能力。

因此實驗中以成就測驗的分數評估學生學習基本程式概念的學習成效，依成就測驗類型分主要分成紙筆測驗與程式實作。依時間分則包含形成性測驗與總結性測驗。測驗內容詳述如後 (測驗內容詳見附錄四、五)：

一、形成性測驗

(1) 紙筆測驗：主要是測試學生在施行策略的當下，對於課程內容的即時性反應。

(2) 程式實作：學生於該堂課下課前，繳交單元程式實作於電腦教室內的作業繳交系統，此部分給予每組的完成時間均相同，主要是在收集學生在實作上的即時學習成效。

二、總結性測驗

(1) 紙筆測驗：包含順序平行、重複執行、條件執行三項基本程式概念綜合試題。

(2) 程式實作：電流急急棒遊戲設計完成。測驗題目雙向細目表如表 8 所示，而表 9 則列出測驗的配分方式。

表 8 測驗題目分配雙向細目表

編號	學習目標	第一單元 順序平行	第二單元 重複執行	第三單元 條件執行
1-1	能瞭解平行/順序的執行方式	1-1-x		
1-2	能預測平行/順序在程式中的運作結果	1-2- (3)		
1-3	能分辨平行與順序執行的不同	1-3- (1) 1-3- (2)		
2-1	能瞭解 repeat 與 forever 的使用方式		2-1-x 2-1- (8) 2-1- (9)	
2-2	能利用重複執行指令，精簡程式		2-2- (7)	
2-3	能預測重複執行指令在程式中的運作結果		2-3- (6)	
3-1	能瞭解 if 與 if-else 的使用方式			3-1-x
3-2	能預測條件執行在程式中的運作結果			3-2- (4) 3-2- (5) 3-2- (10) 3-2- (11)
小結	紙筆測驗總題數=11 題，程式實作 3 題	3	4	4

註：代碼說明 1-1 (目標編號) -1 (測驗題號；程式實作以 x 表示)

表 9 測驗題目類型與配分

題目類型	測驗內容	題數	配分
紙筆測驗	<u>基本程式綜合概念</u> 包含順序平行、重複、條件三單元	11 題	每題 1 分，共 11 分
程式實作	<u>電流急急棒遊戲程式實作</u> 順序平行 2 分（角色說規則） 重複執行 3 分（遊戲障礙物設計） 條件執行 3 分（遊戲規則判斷）	3 題	總和 8 分

測驗的信度與效度，以上學期學過 Scratch 的六年級學生為預試對象，隨機選取 42 人進行預測，所得 Cronbach α 係數為.662<.7，但仍>.5，保有一定的內部一致性信度。測驗效度採專家效度，經學科專家審閱後修正。

試題難度與鑑別度如表 10，其數據顯示測驗題目難度中偏易平均為.716，而鑑別度平均有.477>.40，表試題均保持良好的鑑別度。

表 10 測驗試題難度與鑑別度

題號	難度 (P)	鑑別度 (D)
1	0.750	0.500
2	0.542	0.583
3	0.719	0.562
4	0.553	0.729
5	0.709	0.417
6	0.636	0.395
7	0.875	0.250
8	0.782	0.437
9	0.813	0.375
10	0.813	0.375
11	0.688	0.625
平均	0.716	0.477

4.4.3 學習態度問卷

研究所使用的學習態度問卷分別參考自楊書銘(2008)針對國小六年級學生所自行編纂的 Scratch 學習態度量表，其目的主要是瞭解學生學習 Scratch 的學習態度與感受，問卷以四個面向喜好、焦慮、信心、有用來進行調查，四個面向與全部題目的內部一致性 Cronbach α 係數分別為：.703、.652、.897、.808、.802。而鍾靜宜(2003)針對高中生所設計的學習態度問卷，其目的主要在調查學生對學習迴圈結構教學活動的看法，問卷以學生對於課程內容與教師教學方式的接受度、容易度以及幫助程度三方面進行調查，問卷三個方向與全部內容經內部一致性信度的考驗其 Cronbach α 係數分別為.85、.76、.79、.88，此兩份問卷均有良好的內部一致性信度。

而在本研究中，為了瞭解學生對於此次教學活動的看法，並針對國小四年級學生以及此次的教學主題—控制結構，參考上述兩份問卷進行修改，編製此次研究所使用的學生學習態度問卷其詳細內容如附錄六、七、八。

問卷內容分為四個向度：(1) 情意態度：其中又分為有趣性、愉悅性，主要調查學生對於學習程式設計的實際感受。(2) 學生學習動機：此項目主要調查學生經過此次實驗教學對程式設計的學習動機為何。(3) 各單元難易度：針對此次的三個子單元分別調查學生學習上對單元的難易感受。(4) 不同策略在學生學習上的幫助程度：主要調查學生對於教師所進行的教學策略是否感到有所幫助。

各向度題目分配如表 11，問卷總共 10 題，採 Likert Scale 的四點量尺，1 表很不同意、2 表不同意、3 表同意、4 表非常同意，評分方式為勾選 1 者給 1 分、勾選 2 者給 2 分，以此類推。問卷效度採專家效度，經由專家審閱給予建議後修正。

表 11 學生學習態度問卷向度及題目分配表

問卷向度	子向度	題數	題目編號
情意態度	有趣性	1	1
	愉悅性	1	2
學習動機	學習動機	2	3、4
各單元難易度	順序平行	1	8
	重複執行	1	9
	條件執行	1	10
不同策略在學生學習上的幫助程度 (此部份各組問題不同)	問題導向組	3	5、6、7
	一般教學組	3	5、6、7
	完成問題組	3	5、6、7

4.5 實驗程序

實驗進行為期十一週，利用每週一節資訊課（40 分鐘），實驗階段分為「基本指令教學」與「教學實驗」兩階段。圖 13 表此次教學實驗的實驗流程圖，內容細節說明如後：

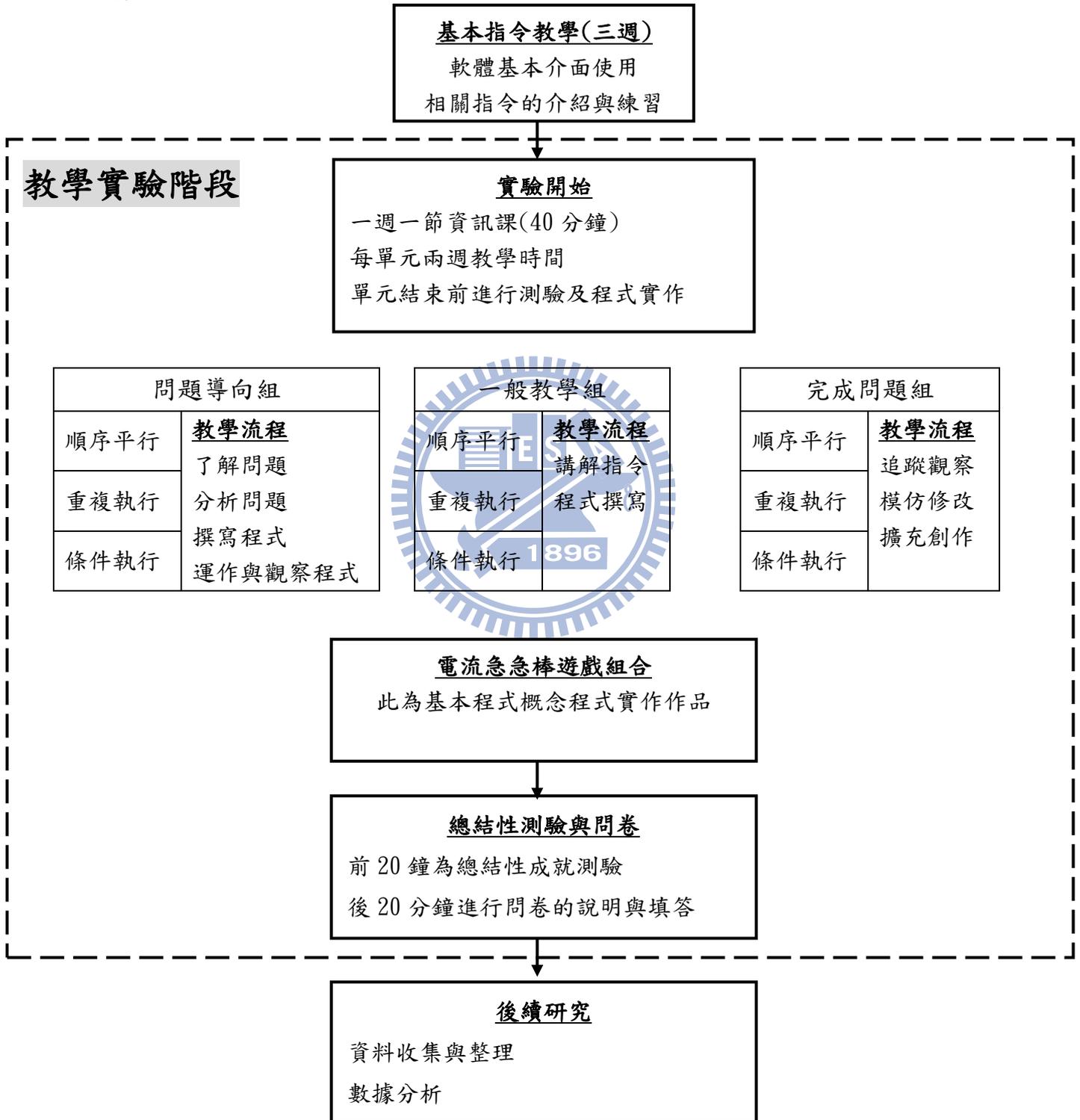


圖 13 實驗程序流程圖

在基本指令教學階段，三組施行一樣的教學課程，主要是因為此次實驗對象為國小四年級學生，對於程式完全沒有接觸過，因此利用三週的時間教導軟體基本介面使用及相關指令的介紹與練習。主要教學內容以週次為單位說明如後，相關教材詳見附錄三：

第一週以 Scratch 基本平台操作為主要教學內容，包括角色（造型、聲音）匯入、指令拖曳、指令執行、儲存檔案等。第二、三週則進行 Scratch 基本指令練習，包括動作、外觀、聲音三個區塊（blocks palette）、以及介紹區塊的堆疊方式等。

而教學實驗階段由第四週之後開始，各班分別進行該組相關的策略活動，每兩週進行一次形成性測驗，主要在收集學生經過該教學策略的單元學習成效，並在課程結束前學生要繳交程式實作任務。在第十週進行電流急急棒的遊戲組合完成，第十一週前 20 分鐘為總結性成就測驗，後 20 分鐘進行問卷填答。將實驗週次整理如表 12 所示：

表 12 教學實驗週次表

週次	教學內容	備註
第一週	Scratch 基本平台操作	三組此時無分別
第二週	Scratch 基本指令練習（一）	三組此時無分別
第三週	Scratch 基本指令練習（二）	三組此時無分別
以上為基本指令教學↑ 以下各組分別施以不同的教學策略↓		
第四週	順序平行	
第五週	順序平行	紙筆測驗與程式實作
第六週	重複執行	
第七週	重複執行	紙筆測驗與程式實作
第八週	條件執行	
第九週	條件執行	紙筆測驗與程式實作
第十週	電流急急棒的遊戲組合	程式實作
第十一週	總結性成就測驗與問卷	總結性成就測驗與 學生學習態度問卷

4.6 資料處理與分析

研究中所搜集之實驗數據，以 SPSS 為主要統計工具，於實驗後，彙整學生的相關測驗成績與學習態度問卷進行量化分析，針對學習成效採單因子變異數分析，而學習態度問卷則以描述性統計進行分析，分析步驟如下，細項如表 13 所示：

(1) 變異數同質性檢定

均先以 Levene 法進行變異數同質性 (homogeneity of variance) 檢定，未達顯著水準，故不違背假設，接著進行變異數分析。

(2) 單因子變異數分析

主要探討不同教學策略下，對學生各項成就測驗的影響，分析細項見表 13。分析結果如達顯著水準，表至少一組有顯著差異，則再繼續進行事後比較。

(3) 事後比較分析

以 Scheffe 法進行事後比較分析，主要用以瞭解三組之間學生學習成效的差異情形。

表 13 數據分析細項表

研究問題	數據來源	統計方法
學生基本程式概念的學習成效是否因不同的教學策略而有所差異？	總結性測驗、 形成性測驗	單因子變異數分析
針對不同的學習單元，分別運用哪種策略其學習成效較佳？	各單元形成性測驗、不同測驗類型測驗成績 (紙筆、實作)	單因子變異數分析
針對高能力組(中能力組、低能力組)的學生，以不同教學策略進行教學，其學習成效上是否有所差異？	總結性測驗	單因子變異數分析
在不同的教學策略下，學生的學習態度為何？	學生學習態度問卷	描述性統計

五、結果與討論

本章主要就三種教學策略（問題導向、一般程式、完成問題）對基本程式概念的學習成效進行分析。採單因子變異數分析（One-Way ANOVA），主要在考驗三組在依變項平均數的差異值是否到統計上的顯著。

5.1 節為學習成效結果分析，將實驗所得各項資料使用 SPSS 統計軟體進行分析。各章節內容分為 5.1.1 各組總結性測驗學習成效分析 5.1.2 各組形成性測驗學習成效分析。而成就測驗類型分為紙筆測驗與程式實作，因此在 5.1.3 小節中探討各組在紙筆測驗類型的學習成效分析，5.1.4 各組在程式實作的學習成效分析。

以學生上學期電腦科成績，進行電腦基本能力分組，欲探討不同基本能力的學生，在不同的教學策略下，對於學習基本程式概念學習成效的影響。將統計結果分節探討 5.1.5 高能力組中，不同教學策略對基本程式概念學習成效之分析。 5.1.6 中能力組中，不同教學策略對基本程式概念學習成效之分析。5.1.7 低能力組中，不同教學策略對基本程式概念學習成效之分析。

本章的 5.2 節針對學生學習態度問卷利用描述性統計進行分析整理。

5.1 學習成效結果分析

5.1.1 各組總結性測驗學習成效分析

如表 14 所示，完成問題組平均數高於其他兩組，為瞭解不同的教學策略對學習成效是否有影響，先以 Levene 法進行變異數同質性檢定，檢定值為 2.697， $p=.073>.05$ ，未達顯著水準，故不違背變異數同質性假設，接著進行變異數分析。

表 14 各組總結性測驗成績描述性統計量

分組	平均數	標準差	標準誤
問題導向組	13.00	3.501	.619
一般教學組	13.77	2.991	.546
完成問題組	14.93	2.203	.409
總和	13.87	3.041	.319

表 15 為三組進行變異數分析，其 F 值=3.246 ($p < .05$)，達到顯著水準，代表至少有一組平均數有顯著差異，需進行事後比較。

表 15 各組總結性測驗成績變異數分析摘要

	平方和	自由度	平均平方和	F 檢定	顯著性
組間	57.189	2	28.594	3.246	.044*
組內	775.229	88	8.809		
總和	832.418	90			

*p<.05

表 16 顯示在完成問題組與問題導向組中，平均差異值=1.931， $p<.05$ ，也就是完成問題組與問題導向組在學習成效上有顯著差異。由之平均數可知，完成問題組學習成效（平均數14.93）優於問題導向組（平均數13.00）。

表 16 各組總結性測驗成績事後比較分析摘要

(I) 組別	(J) 組別	平均差異 (I-J)	標準誤	顯著性
一般教學	問題導向	.767	.754	.598
	完成問題	-1.164	.773	.326
完成問題	問題導向	1.931*	.761	.045
	一般教學	1.164	.773	.326

*p<.05

5.1.2 各組形成性測驗學習成效分析

本小節以各單元形成性測驗成績為數據，共三個成績，分為順序平行單元、重複執行單元、條件執行單元，以下為統計結果：

一、順序平行單元

如表 17 所示，完成問題組平均數高於其他兩組，為瞭解不同的教學策略對順序平行單元的學習成效是否有影響，先以 Levene 法進行變異數同質性檢定，檢定值為 0.015， $p=.986>.05$ ，未達顯著水準，符合變異數同質性假設，接著進行變異數分析。

表 17 各組在順序平行單元成績描述性統計量

分組	個數	平均數	標準差	標準誤
問題導向	32	5.88	1.519	.268
一般教學	30	6.23	1.501	.274
完成問題	29	6.69	1.442	.268
總和	91	6.25	1.510	.158

表 18 為三組進行變異數分析，其 F 值=2.281 ($p>.05$)，未達到顯著水準，表三組在此單元的學習成效並無顯著差異。

表 18 各組在順序平行單元成績變異數分析摘要

	平方和	自由度	平均平方和	F 檢定	顯著性
組間	10.113	2	5.057	2.281	.108
組內	195.074	88	2.217		
總和	205.187	90			

*p<.05

二、重複執行單元

表 19 所示，完成問題組平均數高於其他兩組，為瞭解不同的教學策略對重複執行單元學習成效是否有影響，先以 Levene 法進行變異數同質性檢定，檢定值為 0.538， $p = .568 > .05$ ，未達顯著水準，故符合變異數同質性假設，接著進行變異數分析。

表 19 各組在重複執行單元成績描述性統計量

分組	個數	平均數	標準差	標準誤
問題導向	32	6.22	1.963	.347
一般教學	30	6.73	1.721	.314
完成問題	29	7.21	1.473	.274
總和	91	6.70	1.767	.185

表 20 為三組進行變異數分析，其 F 值=2.463 ($p > .05$)，未達到顯著水準，表三組在此單元的學習成效並無顯著差異。

表 20 各組在重複執行單元成績變異數分析摘要

	平方和	自由度	平均平方和	F 檢定	顯著性
組間	14.895	2	7.447	2.463	.091
組內	266.094	88	3.024		
總和	280.989	90			

*p<.05

三、條件執行單元

表 21 所示，完成問題組平均數高於其他兩組，為瞭解不同的教學策略對條件執行單元學習成效是否有影響，先以 Levene 法進行變異數同質性檢定，檢定值為 2.396， $p = .097 > .05$ ，未達顯著水準，故符合變異數同質性假設，接著進行變異數分析。

表 21 各組在條件執行單元成績描述性統計量

分組	個數	平均數	標準差	標準誤
問題導向	32	7.50	2.676	.473
一般教學	30	8.40	2.343	.428
完成問題	29	9.66	1.653	.307
總和	91	8.48	2.424	.254

表 22 為三組進行單因子變異數分析，F 值=6.822 ($p > .05$)，達到顯著水準，代表至少有一組平均數有顯著差異，需進行事後比較。

表 22 各組「條件執行單元成績變異數分析摘要

	平方和	自由度	平均平方和	F 檢定	顯著性
組間	70.974	2	35.487	6.822	.002*
組內	457.752	88	5.202		
總和	528.725	90			

* $p < .05$

表 23 完成問題組與問題導向組中，平均差異值=2.155 ($p < .05$)，也就是完成問題組與問題導向組在學習成效上有顯著差異。由表 21 之平均數可知，完成問題組的學習成效（平均數9.66）優於問題導向組（平均數7.50）。

表 23 各組在條件執行單元學習成績進行事後比較分析摘要

(I) 組別	(J) 組別	平均差異 (I-J)	標準誤	顯著性
一般教學	問題導向	.900	.580	.304
	完成問題	-1.255	.594	.113
完成問題	問題導向	2.155*	.585	.002
	一般教學	1.255	.594	.113

* $p < .05$

綜合以上資料分析，完成問題組在總結性測驗上優於問題導向組，但與一般程式教學組相比則無顯著性差異，但細看其平均數則有完成問題高於一般程式教學高於問題導向的趨勢。

而在各單元形成性測驗上，除了條件執行單元上完成問題組與問題導向組有顯著差異，且完成問題組之條件單元學習成效優於問題導向組，其他兩個單元均無差異。而細看各單元之平均數，皆有完成問題高於一般程式教學高於問題導向的趨勢。

5.1.3 各組紙筆測驗學習成效分析

以 Oliver 在程式知識上的分類，在此節中將紙筆測驗歸類於程式設計的概念性知識，以下將探討各組在紙筆測驗上的學習成效差異。

由表 24 中可看出，在所有的紙筆測驗上，其平均數也均以完成問題組高為最高。以下測驗成績均先以 Levene 法進行變異數同質性檢定，數據均未達顯著水準，故不違背變異數同質性假設，接著進行變異數分析。

表 24 各組在紙筆測驗上的描述性統計量

紙筆測驗	組別	個數	平均數	標準差	標準誤
順序平行	問題導向	32	4.13	1.070	.189
	一般教學	30	4.33	1.061	.194
	完成問題	29	4.41	1.211	.225
	總和	91	4.29	1.108	.116
重複執行	問題導向	32	3.69	1.615	.286
	一般教學	30	4.60	1.380	.252
	完成問題	29	4.66	1.289	.239
	總和	91	4.30	1.494	.157
條件執行	問題導向	32	6.13	2.211	.391
	一般教學	30	6.67	1.918	.350
	完成問題	29	7.83	1.284	.238
	總和	91	6.85	1.972	.207
總結性測驗	問題導向	32	6.94	2.526	.447
	一般教學	30	6.97	2.297	.419
	完成問題	29	7.86	2.150	.399
	總和	91	7.24	2.349	.246

表 25 為各組在紙筆測驗上的變異數分析總表。重複執行單元，F值=4.424 ($p<.05$) 以及條件執行單元，F值=6.585 ($p<.05$)，達到顯著水準，代表至少有一組平均數有顯著差異，需進行事後比較。

表 25 各組紙筆測驗變異數分析摘要總表 (續)

紙筆測驗		平方和	自由度	平均平方和	F 檢定	顯著性
順序平行	組間	1.370	2	.685	.552	.578
	組內	109.201	88	1.241		
	總和	110.571	90			
重複執行	組間	18.362	2	9.181	4.424	.015*
	組內	182.627	88	2.075		
	總和	200.989	90			

條件執行	組間	45.542	2	22.771	6.585	.002*
	組內	304.305	88	3.458		
	總和	349.846	90			
總結性測驗	組間	16.391	2	8.196	1.502	.228
	組內	480.290	88	5.458		
	總和	496.681	90			

*p<.05

表 26 為重複執行及條件執行在紙筆測驗上的事後分析比較總表。以重複執行單元來看，完成問題組在紙筆測驗的學習成效與問題導向組有顯著差異（平均差異=0.968， $p=.037<.05$ ）。而與一般教學組與問題導向組中也有顯著差異（平均差異=0.912， $p=.050<.05$ ）。由表 24 之平均數可知，完成問題組的重複執行單元學習成效（平均數4.66）表現比問題導向組（平均數3.69）學習成效好。而一般教學組重複執行單元學習成效（平均數4.60）表現比問題導向組（平均數3.69）學習成效好。

以條件執行單元來看，完成問題組在紙筆測驗的學習成效與問題導向組有顯著差異（平均差異=1.703， $p=.003<.05$ ）。由表 24 之平均數可知，完成問題組的條件執行單元學習成效（平均數7.83）表現比問題導向組（平均數6.13）學習成效好。

表 26 各組在重複執行及條件執行紙筆測驗上事後比較分析摘要總表

依變數	(I) 組別	(J) 組別	平均差異 (I-J)	標準誤	顯著性
重複執行	一般教學	問題導向	.912*	.366	.050
		完成問題	-.055	.375	.989
	完成問題	問題導向	.968*	.369	.037
		一般教學	.055	.375	.989
條件執行	一般教學	問題導向	.542	.473	.521
		完成問題	-1.161	.484	.062
	完成問題	問題導向	1.703*	.477	.003
		一般教學	1.161	.484	.062

*p<.05

5.1.4 各組程式實作學習成效分析

以 Oliver 在程式知識上的分類，在此節中將程式實作歸類於策略性知識，以下將探討各組程式實作上的學習成效差異。

由表 27 中可看出，在所有的程式實作上，其平均數也均以完成問題組高為最高。以下測驗成績均先以 Levene 法進行變異數同質性檢定，數據均未達顯著水準，故不違背變異數同質性假設，接著進行變異數分析。

表 27 各組程式實作的描述性統計量

程式實作	分組	個數	平均數	標準差	標準誤
順序平行	問題導向	32	1.75	.916	.162
	一般教學	30	1.90	.712	.130
	完成問題	29	2.28	.751	.139
	總和	91	1.97	.823	.086
重複執行	問題導向	32	2.53	.671	.119
	一般教學	30	2.13	.776	.142
	完成問題	29	2.55	.632	.117
	總和	91	2.41	.715	.075
條件執行	問題導向	32	1.38	1.040	.184
	一般教學	30	1.73	1.081	.197
	完成問題	29	1.83	.966	.179
	總和	91	1.64	1.038	.109
電流急急棒遊戲	問題導向	32	6.06	1.777	.314
	一般教學	30	6.80	1.424	.260
	完成問題	29	7.07	1.100	.204
	總和	91	6.63	1.518	.159

表 28 為各組在程式實作測驗上的變異數分析總表，在順序平行程式實作，F 值 = 3.433 ($p < .05$) 以及遊戲程式實作上，F 值 = 3.870 ($p < .05$)，達到顯著水準，代表至少有一組平均數有顯著差異，需進行事後比較。

表 28 各組程式實作測驗上的變異數分析摘要總表 (續)

程式實作		平方和	自由度	平均平方和	F 檢定	顯著性
順序平行	組間	4.408	2	2.204	3.433	.037*
	組內	56.493	88	.642		
	總和	60.901	90			
重複執行	組間	3.348	2	1.674	3.458	.036
	組內	42.608	88	.484		
	總和	45.956	90			

條件執行	組間	3.528	2	1.764	1.660	.196
	組內	93.505	88	1.063		
	總和	97.033	90			
電流急急棒遊戲	組間	16.760	2	8.380	3.870	.024*
	組內	190.537	88	2.165		
	總和	207.297	90			

*p<.05

表 29 為順序平行程式實作及遊戲程式實作的事後分析比較總表。以順序平行程式實作來看，完成問題組的學習成效與問題導向組有顯著差異（平均差異=0.526， $p<.05$ ）。由表 27 之平均數可知，完成問題組的順序平行程式實作學習成效（平均數 2.28）優於問題導向組（平均數 1.75）。

以遊戲程式實作來看，完成問題組的學習成效與問題導向組有顯著差異（平均差異=1.006， $p<.05$ ）。由表 27 之平均數可知，完成問題組的條件程式實作學習成效（平均數 7.07）優於問題導向組（平均數 6.06）。

表 29 各組在順序平行及遊戲程式實作事後比較分析摘要總表

依變數	(I) 組別	(J) 組別	平均差異 (I-J)	標準誤	顯著性
順序平行	一般教學	問題導向	.150	.204	.763
		完成問題	-.376	.209	.203
	完成問題	問題導向	.526*	.205	.042
		一般教學	.376	.209	.203
電流急急棒 遊戲	一般教學	問題導向	.737	.374	.149
		完成問題	-.269	.383	.782
	完成問題	問題導向	1.006*	.377	.033
		一般教學	.269	.383	.782

*p<.05

5.1.5 高能力組中，不同教學策略對基本程式概念學習成效之分析

以學生上學期電腦課平均成績，抽取出各班前 25% 為高能力組進行比較，如表 30 中所示，完成問題組平均數高於其他兩組，為瞭解在高能力組中不同的教學策略對學習成效是否有影響，先以 Levene 法進行檢定，檢定值為 1.151 ($p=.337>.05$)，未達顯著水準，故符合變異數同質性假設，接著進行變異數分析。

表 30 高能力組學生在總結性測驗描述性統計量

分組	個數	平均數	標準差	標準誤
問題導向	8	13.50	3.780	1.336
一般教學	7	15.29	2.498	.944
完成問題	7	14.86	2.795	1.056
總和	22	14.50	3.067	.654

表 31 為在高能力組以三種教學策略為自變項，針對總結性成績進行變異數分析，其 F 值=0.681，($p=.518>.05$)，未達到顯著水準。

表 31 高能力組中，不同教學策略在總結性測驗成績變異數分析摘要

	平方和	自由度	平均平方和	F 檢定	顯著性
組間	13.214	2	6.607	.681	.518
組內	184.286	19	9.699		
總和	197.500	21			

* $p<.05$

5.1.6 中能力組中，不同教學策略對基本程式概念學習成效之分析

以學生上學期電腦課平均成績，抽取出各班中間 50% 為中能力組進行比較，如表 32 中所示，完成問題組平均數高於其他兩組，為瞭解在中能力組在不同的教學策略對學習成效是否有影響，先以 Levene 法進行檢定，檢定值為 2.107，($p=.135>.05$)，未達顯著水準，故符合變異數同質性假設，接著進行變異數分析。

表 32 中能力組在總結性測驗成績描述性統計量

分組	個數	平均數	標準差	標準誤
問題導向	15	14.73	1.870	.483
一般教學	13	13.62	2.815	.781
完成問題	14	14.93	1.979	.529
總和	42	14.45	2.255	.348

表 33 在中能力組以三種教學策略為自變項，針對總結性測驗成績進行變異數分析，其 F 值=1.347 ($p>.05$)，未達到顯著水準。

表 33 中能力組中，不同教學策略在總結性測驗成績變異數分析摘要

	平方和	自由度	平均平方和	F 檢定	顯著性
組間	13.466	2	6.733	1.347	.272
組內	194.939	39	4.998		
總和	208.405	41			

*p<.05

5.1.7 低能力組中，不同教學策略對基本程式概念學習成效之分析

以學生上學期電腦課平均成績，抽取出各班後 25% 為低能力組進行比較，如表 34 中所示，完成問題組平均數高於其他兩組，為瞭解在低能力組中不同的教學策略對學習成效是否有影響，先以 Levene 法進行檢定，檢定值為 1.152， $p = 0.333 > 0.05$ ，未達顯著水準，故符合變異數同質性假設，接著進行變異數分析。

表 34 低能力組學生在總結性測驗成績描述性統計量

分組	個數	平均數	標準差	標準誤
問題導向	9	9.67	3.240	1.080
一般教學	10	12.90	3.381	1.069
完成問題	8	15.00	2.330	.824
總和	27	12.44	3.662	.705

表 35 為以三種教學策略為自變項，針對總結性測驗成績進行變異數分析，F 值 = 6.604 ($p < .05$)，達到顯著水準，代表至少有一組平均數有顯著差異，需進行事後比較。

表 35 低能力組中，不同教學策略總結性測驗成績變異數分析摘要

	平方和	自由度	平均平方和	F 檢定	顯著性
組間	123.767	2	61.883	6.604	.005*
組內	224.900	24	9.371		
總和	348.667	26			

*p<.05

表36 為低能力組中，以不同教學策略的總結性測驗成績的事後分析比較表，以完成問題組與問題導向組來看，平均差異值=5.333 ($p < .05$)，也就是在低能力學生組當中，完成問題組與問題導向組在學習成效上有顯著差異。由表 34 之平均數可知，低能力學生中，以完成問題組的學習成效（平均數15）優於問題導向組（平均數9.67）。

表36低能力組中，不同教學策略在總結性測驗成績的事後分析比較表

(I) 組別	(J) 組別	平均差異 (I-J)	標準誤	顯著性
一般教學	問題導向	3.233	1.407	.092
	完成問題	-2.100	1.452	.367
完成問題	問題導向	5.333*	1.487	.006
	一般教學	2.100	1.452	.367

*p<.05

5.1.8 小結

分析結果以表 37 列出各組在總結性測驗、形成性測驗以及紙筆測驗與程式實作測驗上的統計結果。結果顯示在學習成效上多以完成問題優於問題導向，將結果分述如下：

- (1) 總結性測驗：完成問題優於問題導向。
- (2) 形成性測驗：在條件執行單元完，完成問題優於問題導向。
- (3) 紙筆測驗：重複執行及條件執行單元，完成問題優於問題導向。且在重複執行單元中，一般教學組也優於問題導向組。
- (4) 程式實作：順序平行及遊戲程式實作，完成問題優於問題導向。

表 37 學習成效分析總表

學習成效	問題導向組	一般教學組	完成問題組
總結性測驗	—	—	優於問題導向
- 紙筆測驗	—	—	—
- 程式實作	—	—	優於問題導向
順序平行單元	—	—	—
- 紙筆測驗	—	—	—
- 程式實作	—	—	優於問題導向
重複執行單元	—	—	—
- 紙筆測驗	—	優於問題導向	優於問題導向
- 程式實作	—	—	—
條件執行單元	—	—	優於問題導向
- 紙筆測驗	—	—	優於問題導向
- 程式實作	—	—	—

註：—表示統計上無顯著差異。

而以上學期電腦課成績進行電腦基本能力分組，高能力組及中能力組，以不同的教學策略進行教學在教學成效尚無顯著差異，但對於低能力組的學生，完成問題組別優於問題導向組別。整理如表 38 所示：

表 38 不同能力分組學生的學習成效分析總表

不同電腦基本能力分組	低能力組	中能力組	高能力組
總結性測驗成績	有顯著差異， 且完成問題組優 於問題導向組	無顯著	無顯著

5.2 學生學習態度問卷分析結果

此份問卷主要針對三組學生的學習態度進行調查，問卷內容分四個向度，以下利用各小節則要說明分析結果，各小節內容為：5.2.1 針對學生情意態度進行探討。5.2.2 針對學生學習動機進行探討。5.2.3 針對各單元難易度進行探討。5.2.4 分別探討每組學生對教師設計的教學策略，在學習上的幫助程度的感受。而問卷採 Likert Scale 的四點量尺，1 代表很不同意、2 代表不同意、3 代表同意、4 代表非常同意，分數越高者表越同意。

5.2.1 學生情意態度

一、有趣性

表 39 顯示三種教學策略下，學生對於學習程式設計是否覺得有趣。因為學生在進行教學實驗之前，資訊課的課程內容為文書處理軟體（PowerPoint）因此將程式設計課程拿來與一般應用軟體課程相比較，結果顯示五成的學生（非常同意）明顯認為程式設計課程比較有趣。而又以問題導向組別比例較高。

表 39 學生對程式設計課程是否覺得有趣調查結果

量尺	非常同意	同意	不同意	很不同意
組 問題導向組	59.4%	31.3%	3.1%	6.3%
別 一般教學組	43.3%	43.3%	3.3%	10%
完成問題組	51.7%	37.9%	3.5%	6.9%

二、愉悅性

表 40 與 表 41 皆在調查學生對於程式設計課程的愉悅感，幾乎八成以上的學生（含非常同意與同意）對於學習程式設計皆感到愉快。

表 40 學生對學習程式設計課程是否感到愉悅調查結果一

量尺	非常同意	同意	不同意	很不同意
組 問題導向組	65.6%	21.9%	3.1%	9.4%
別 一般教學組	43.3%	46.7%	0%	10%
完成問題組	55.2%	37.9%	3.4%	3.4%

表 41 學生對學習程式設計課程是否感到愉悅調查結果二

量尺	非常同意	同意	不同意	很不同意
組 問題導向組	68.8%	21.9%	3.1%	6.3%
別 一般教學組	46.7%	33.3%	13.3%	6.7%
完成問題組	69.0%	20.7%	6.9%	3.4%

5.2.2 針對學生學習動機進行探討

表 42 與 表 43 針對學生未來是否願意主動學習程式設計來進行探討，全體學生近九成（含非常同意與同意）皆表示有興趣及未來願意再繼續進行創作。

表 42 學生學習動機調查結果一

量尺	非常同意	同意	不同意	很不同意
組 問題導向組	59.4%	31.3%	3.1%	6.3%
別 一般教學組	46.7%	40%	3.3%	10%
完成問題組	58.6%	34.5%	3.4%	3.4%

表 43 學生學習動機調查結果二

量尺	非常同意	同意	不同意	很不同意
組 問題導向組	59.4%	28.1%	3.1%	9.4%
別 一般教學組	53.3%	36.7%	0%	10%
完成問題組	58.6%	34.5%	3.4%	3.4%

5.2.3 針對各單元難易度進行探討

表 44、表 45、表 46 分別針對各單元困難度的調查結果進行分析。

表 44 順序平行單元難易度調查結果

量尺	很容易	容易	有點難	非常困難
組 問題導向組	40.6%	34.4%	18.8%	6.3%
別 一般教學組	46.7%	30%	16.7%	6.7%
完成問題組	65.5%	31%	0%	3.4%

表 45 重複執行單元難易度調查結果

量尺	很容易	容易	有點難	非常困難
組 問題導向組	62.5%	31.3%	0%	6.3%
別 一般教學組	56.7%	30%	6.7%	6.7%
完成問題組	51.7%	41.4%	3.4%	3.4%

表 46 條件執行單元難易度調查結果

量尺	很容易	容易	有點難	非常困難
組 問題導向組	37.5%	28.1%	28.1%	6.3%
別 一般教學組	40%	40%	6.7%	13.3%
完成問題組	55.2%	24.1%	13.8%	6.9%

綜合以上數據顯示，學生認為學習上有困難的單元以條件執行單元（非常困難 26.5%）為最多，而三組學生認為比較簡單的為重複執行單元（三組在很容易選項中均有五成學生）。

5.2.4 不同策略對學生學習上的幫助程度

探討每組學生對教師設計的教學策略，在學習上的幫助程度的感受。

如表 47 所示，問題導向組的學生對於教師一步步引導對問題的瞭解與分析是有幫助的（至少有 70% 以上的學生為非常同意），但對於瞭解基本程式設計概念，則只有六成學生表示有幫助。

表 47 問題導向策略對學生學習上的幫助程度

量尺	非常同意	同意	不同意	很不同意
策 瞭解問題	71.9%	21.9%	0%	6.3%
略 分析問題	81.3%	12.5%	0%	6.3%
別 整體感受	62.5%	34.4%	0%	3.1%

註：問卷題目（以下三題對應問題導向策略教學步驟）

透過老師分析角色的動作，讓我瞭解到基本程式設計的概念（瞭解問題）

透過老師上課前，一步步分析要怎麼完成任務，讓我更容易完成我的遊戲（分析問題）

透過這次課程，我瞭解到基本的程式設計概念（整體感受）

如表 48 所示，一般教學組的學生，有六成（非常同意）認為這樣的教學方法是有幫助的，而對於瞭解基本程式設計概念，也是六成的學生表示有幫助。

表 48 一般教學策略對學生學習上的幫助程度

量尺	非常同意	同意	不同意	很不同意
策 指令說明	66.7%	23.3%	3.3%	6.7%
略 程式實作	63.3%	26.7%	3.3%	6.7%
別 整體感受	60%	26.7%	6.7%	6.7%

註：問卷題目（以下三題對應一般程式教學的教學步驟）

透過老師講解每個指令的用法，讓我瞭解到基本程式設計的概念（指令說明）

透過老師上課前，講解指令的用法，讓我更容易完成我的遊戲（程式實作）

透過這次課程，我瞭解到基本的程式設計概念（整體感受）

如表 49 所示，完成問題組的學生，有七成的學生認為觀察範例是有幫助的，有六成的學生覺得在模仿修改的任務中，對他之後的遊戲完成是有幫助的。而在瞭解基本程式概念上也是有六成的學生認為是有幫助。

表 49 完成問題教學策略對學生學習上的幫助程度

量尺	非常同意	同意	不同意	很不同意
策 追蹤觀察	72.4%	27.6%	0%	0%
略 模仿修改	62.1%	37.9%	0%	0%
別 整體感受	69%	27.6%	3.4%	0%

註：問卷題目（以下三題對應完成問題策略教學步驟）

透過每堂課會先看範例，並完成學習單，讓我瞭解到基本程式設計的概念（追蹤觀察）

透過第一堂課的例子，以及完成老師給我的部分任務，讓我更容易完成我的遊戲（模仿修改）

透過這次課程，我瞭解到基本的程式設計概念（整體感受）



六、結論與建議

本研究主要目的為探討教師在教導基本程式設計時，運用不同的教學策略對學生在學習成效上的影響，並期望研究結果能提供給國小資訊教師在進行教學設計時，參考使用。並深入探討在不同電腦基本能力下，對於該組學生（高能力組、中能力組、低能力組），使用不同的教學策略對其在學習成效上的影響。最後，針對學生的學習動機進行調查分析。

本章總共分為二個部分，6.1 為研究結論，係將第五章之實驗結果加以整理分析，並針對本研究之待答問題進行答覆。6.2 為建議，根據本研究結果與研究者實際進行研究之心得提出建議，以做為教師教學及未來研究者研究設計開發之參考。

6.1 結論

經過實驗教學後，將所收集的數據資料經過整理分析，分別歸納出以下結論：

一、學生學習成效上

根據總結性測驗分析結果來看，學生的整體學習成效，完成問題組與問題導向組有顯著差異，而又以完成問題組學習成效較佳，此研究結果，與 Sweller 等人在（1998）提出的概念相符合，學者們指出，針對像是程式設計這樣的學科領域中，對於初學者利用範例教學會比一開始就利用問題解決進行教學更有效。而對於一般程式教學，在學習成效上則與另兩種策略沒有顯著差異。

根據形成性測驗分析結果來看，只有條件執行單元在完成問題組與問題導向組上是有顯著差異，而又以完成問題組學習成效較佳。配合問卷中的單元難易度調查，學生認為最困難的單元（有高達 26.5%）即為條件執行。但反觀認為條件執行單元很容易的人數，完成問題組有 55.2%、問題導向組 37.5%、一般教學組 40%，其人數則以完成問題組遠高於其他兩組。因此推估，在進行比較難或是複雜的觀念教學時，利用完成問題策略來引導學生學習，對於學生的學習感受以及學習成效上會有較佳的成果。

依據測驗類型，將數據作細部分析，分成紙筆測驗與程式實作兩個部份的學習成效。在紙筆測驗上，以重複及條件執行單元上有顯著差異，且以完成問題優於問題導向，以 Oliver（1993）對程式設計知識的分類，在紙筆測驗上學生針對程式碼進行評估然後作答，此即 Oliver 所提到的概念性知識。因此推估，在較為複雜及困難的單元中，針對程式碼的構造及語法，運用完成策略對於學生的學習成效會有較佳的效果。但在順序平行單元的紙筆測驗上三組並無顯著差異，這與許宏彰（2005）針對國小學生 Logo 語言

程式設計思維歷程之研究中指出學生多採用順向思考的方式進行解題，而在順序平行的單元上，其教學目標為用系統性的思考方式將想要執行的步驟依序執行，此與學生的思考歷程相符，所以在紙筆測驗上並無顯著差異。

在程式實作的分析結果中，條件與重複執行單元上，三組無顯著差異。究其原因，在程式實作的題型中，學生會實際面臨真實的問題情境，學習者除了要具備基本的程式知識外，針對問題也需要完整了解，因此接受問題導向組別的學生在此種題型中，相較於另兩組，或許保有較好的問題解決概念（鍾靜宜，2003；Pedersen & Liu; 2003），因此無顯著差異。但對於順序平行以及遊戲實作的部分，在完成問題組與問題導向組上有顯著差異，又以完成問題組優於問題導向。

二、針對不同電腦基本能力的學生（高能力組、中能力組、低能力組），使用不同的教學策略學習基本程式概念學習成效的影響

在高能力組與中能力組，施以不同的教學策略在學習成效上均無顯著差異，但對於低能力組的學生，以完成問題組與問題導向組上有顯著差異，又以使用完成問題策略學習成效較佳。究其原因，低能力組學生其本身電腦基本操作能力較為不足，因此在學習程式概念課程時，除了要學習新知，還要處理基本操作的思考，這對學生來說會造成學習時的認知負荷過高，而範例觀察搭配完成問題正好可以降低學生學習程式設計的認知負荷（Garner, 2009），這也說明了研究結果，因此未來在針對低能力的學生，可以運用完成問題策略進行初始階段的教學策略。

三、學生的學習態度調查

問卷調查顯示學生對於學習程式設計大多都能樂於接受且勇於嘗試。輔以研究者的觀察，因為學生實際體認到的為遊戲設計，因此對於程式概念的引導多數皆能保持高度的學習動機，並加以學習。

而在學習動機上，經研究者與學生對談後，有學生反應家中無法上網所以沒辦法下載軟體，或是從 Scratch 官方網站進入後為英文介面以及無法找到程式載點，這些考量均會影響到學生在回答此問題的看法。但在學習動機的填答上也有五成的學生樂於繼續學習，此正也是教師在教學時的最重要的目標—引起學生自動自發的學習。

而在各個策略的幫助程度上，有六成學生對於教師在各個教學策略上均認為有所幫助，但在整體感受上，仍以完成問題組的學生 69% 為最高，更有 72.4% 的學生肯定範例觀察對他們的幫助，可是在完成問題（模仿修改策略）上，有 62.1% 的學生認為是有幫助的，推估原因，經由研究者觀察發現學生對於遊戲的設計都保持高度興趣，很想在課堂上馬上學以致用，但是教師所提供的模仿修改策略主要還是針對基本程式概念的引

導，與第三部份擴充創作（遊戲製作）的任務會有所不同，因此學生在進行活動二時，會造成任務完成率不高，有學生會直接進行活動三的程式實作。因此，教師未來在設計相關活動時，可以針對此部分加以考量。

6.2 未來研究方向

此小節內容包含研究結束後，研究者對於此次實驗實施過程的相關缺失、限制與心得，在此提出以提供給未來研究相同領域的研究者進行參考。分別就教材設計、實際教學教學現場、未來發展等三部份，提出以下幾點建議。

一、完成問題策略教材設計

就如同 Garner (2003) 說到產出一個好的範例程式，讓學生可以擴充修改是一個很耗費時間的過程。這也是此次在實驗過程中最耗費時間與心力的地方，教師在利用步驟向下減少的技巧設計教材時，除了要考量到學生的認知負荷還必須將欲傳達的概念融入在完成問題策略的步驟中，這其實是要透過多次的討論以及整合眾多教學專家的意見。因此，雖然研究結果顯示完成問題學習成效優於問題導向，但在實際教學現場上，一般資訊教師要設計良好的範例程式以及完成問題範例，是很耗時的。

但是摒除教材設計的困難點，三種策略均有其優缺點，教師可以多加利用完成問題策略的相關概念，與一般教學、問題導向教學相互融合，進行混合式的策略應用，這對於學生在理解較為困難的程式概念時（像是條件、變數、隨機等），必定有所助益。

二、研究限制

研究中教學者與施測者為同一人，因此教師在教學時，自身需要將三種策略做清楚的切割，才不會影響到實驗結果，而在設計成就測驗時，也必須公平的考量三組學生所使用到的程式指令與範例。因此，在未來進行相關的研究時，建議可將施測者與教學者分割，在學習成效的測量上應會較為客觀。

三、實際教學現場

以研究者從旁觀察，學生對於範例的觀察輔以學習單的填寫都有高度的動機，但此次教學實驗的主軸為遊戲設計，當初立意是為增進學生動機，但在這樣的教學主題下，學生在活動二的完成問題，就顯得沒那麼專注，學生會急著想進行遊戲三的遊戲創作。再者，此次的程式設計軟體使用的是 Scratch，其豐富的角色圖庫以及類似小畫家的繪圖介面，會讓學生的注意力分散，雖然學生最後的實作成品完成度都很高，但仍是建議，

如要使用完成問題策略設計教材，可以搭配文字式的程式設計軟體，像是 Drape，讓學生可以專注在初階程式概念的學習。

四、未來發展

1、製作動態教學輔具

因此次的實驗重點在策略的設計，未來可以搭配 Scratch 相關硬體（PicoBoard http://info.scratch.mit.edu/Sensor_Boards）讓程式與現實生活相連接，讓範例觀察以及完成問題相關程式視覺化、具體化，以期能促進學生在學習抽象的程式概念時，更能有所覺察。

2、使用它種程式設計語言來進行教學

一般皆認為使用圖像式對學生來說比較容易接受，但經過此次實驗後，發現圖像式的程式設計軟體雖較能引起學生動機，而學生在創意上也有很驚人表現，但對於基本程式概念、邏輯思考能力的養成，在未來可以試著先利用文字式程式語言進行教學，再利用圖像式程式語言進行進階作品創作的相關研究。



參考文獻

一、中文部分

- 王麒富 (2009)。應用直觀式 Scratch 軟體提升國小學生問題解決能力效益之研究。國立臺中教育大學，碩士論文，台中。
- 宋曜廷 (2000)。先前知識、文章結構和多媒體呈現對文章學習的影響。國立台灣師範大學教育心理與輔導學系，博士論文，未出版，台北。
- 李昀龍 (2006)。引導合作學習對於國小學生學習 Logo 程式設計之影響。國立臺灣師範大學資訊教育學系，碩士論文，未出版，台北。
- 林美娟 (2005)。國科會計畫「以實驗策略增進學生的程式設計概念與解題能力」完整報告 (NSC93-2520-S-003-007-)。
- 林宥雯 (2006)。以認知負荷理論探討國小社會教科書課文內容編排方式對國小高年級學生閱讀理解表現之影響。高雄師範大學，碩士論文，高雄。
- 林裕雲 (2002)。實施電腦 LOGO 程式設計教學對台灣國小學生解題能力之影響—國小六年級學生之個案研究。屏東師範學院數理教育研究所，碩士論文，屏東。
- 教育部 (2008)。97 年國民中小學九年一貫課程綱要 (100 學年度實施)。臺北：教育部。
- 張文奇 (2008)。視覺化程式設計對國小兒童高層次思考能力之影響。臺北市立教育大學自然科學教學碩士學位班，碩士論文，未出版，台北。
- 許宏彰 (2005)。國小學生 LOGO 語言程式設計思維歷程之研究。國立臺北教育大學數學教育研究所，碩士論文，台北。
- 陳明溥 (2007)。程式語言課程之教學模式與學習工具對初學者學習成效與學習態度之影響。師大學報，52，1-21。
- 陳蜜桃 (2003)。認知負荷理論及其對教學的啟示。教育學刊，21，29-51。
- 黃文聖 (2001)。國小學生在 Logo 學習環境中數學學習與解題之研究。國立新竹師範學院數理教育研究所。碩士論文，新竹。
- 楊書銘 (2008)。Scratch 程式設計對六年級學生邏輯推理能力、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，碩士論文，未出版，台北。
- 楊培渝 (2006)。認知負荷、焦慮情境對英語閱讀成效之影響。國立雲林科技大學技術及職業教育研究所碩士論文，未出版，雲林縣。
- 劉明洲 (2003)。程式語言學習者對於變數之認知與應用情形研究。第 11 屆國際電腦輔助教學研討會 (ICCAI2003)，國立台灣師範大學。
- 賴健二 (2003)。兒童視覺化中文程式語言之開發與研究。國立臺北師範學院教育傳播與科技研究所，碩士論文，台北。

- 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。發表於 GCCCE2008 第十二屆全球華人計算機教育應用大會。臺北：國立臺灣師範大學資訊教育教育研究所。
- 譚俊彥 (1993)。口述思考配對解題在程式語言教學之應用。國立台灣師範大學資訊教育學系，碩士論文，台北市。
- 鐘靜宜 (2003)，教學策略與學習工具對高中程式語言學習之影響。國立台灣師範大學資訊教育研究所，碩士論文，台北。

二、英文部份

- Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research*, 70, 181-214.
- Atkinson, R. K., Renkl, A., & Merrill, M. M. (2003). Transitioning from studying examples to solving problems: effects of self-explanation prompts and fading worked-out steps. *Journal of Educational Psychology*, 95, 774-783.
- Clark, R.C., Nguyen, F., and Sweller, J. (2005). *Efficiency in Learning Evidence-based guidelines to manage cognitive load*. San Francisco: CA: Pfeiffer.
- Dalbey, J. and Linn, M. C. (1986). Cognitive consequences of programming: Augmentations to basic instruction. *Journal of Educational Computing Research*, 2, 75-93.
- Deimel, L. E. and Moffat, D. V. (1982). A more analytical approach to teaching the introductory programming course. In J. Smith and M. Schuster (Eds.), *Proceedings of the NECC* (pp.114-118). Columbia: The University of Missouri.
- Fain, Y.(2004). Teaching kids programming: Even younger kids can learn Java. Retrieved December 18, 2009, from <http://java.sys-con.com/node/44575>
- Fowler, W. A. L., & Fowler, R. H. (1993). A hypertext approach to computer science education unifying programming principles. *Journal of Multimedia and Hypermedia*, 2(4), 433-441.
- Garner, S.(2003). Learning to program using part-complete solutions. In *Computer Based*

Learning in Science, Nicosia, Cyprus.

Garner, S.(2009). A quantitative study of a software tool that supports a Part-Complete Solution Method on learning outcomes. *Journal of Information Technology Education*, 8,285-

Gomes, A. & Mendes, A.(2008). A study on student's characteristics and programming learning. *In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008(pp. 2895-2904)*. Chesapeake, VA: AACE.

Govender, I. & Grayson, D.(2006). Learning to program and learning to teach programming: A closer look. *Paper presented at the 2006 World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA2006)*, Orlando, Florida.

Kalyuga, S., Ayres, P., Chandler, P., & Sweller, J.(2003). The expertise reversal effect. *Educational Psychologist*, 38, 23-32.

KE Chang, BC Chiao, SW Chen, RS Hsiao.(2000). A programming learning system for Beginners - a completion strategy approach. *IEEE Transactions on Education*, 43(2), 211-220.

Kelleher, C., & Pausch, R.(2005).Lowering the barriers to programming: A Taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*,37 (2), 83-137.

Kirsner, K.(1998). Implicit and explicit mental processes. Mahwah, NJ: LEA.

Moreno, R.(2006).When worked examples don't work: Is cognitive load theory at an impasse? *Learning and Instruction*, 16, 170-181.

Pedersen,S. & Liu,M. (2003).The transfer of problem-solving skills from a problem-based learning environment: The effect of modeling an experts cognitive process. *Journal of Research on Technology in Education*. 35(2), 303-330.

Resnick, M. (2007). Sowing the Seeds for a More Creative Society. Learning and Leading with Technology. Retrieved February 6, 2009, from <http://scratch.wik.is/Research>

- Reichert, R., Nievergelt, J., & Hartmann, W.(2001). Programming in schools-why, and how ? ,
 In C. Pellegrini, A. Jacquesson (Hrsg.): *Enseigner l'informatique*, 143-152.
- Renkl, A., Atkinson, R. K., & Maier, U. H.(2000). From studying examples to solving problems: Fading worked-out solution steps helps learning. *Proceeding of the 22nd Annual Conference of the Cognitive Science Society* (pp. 393–398). Mahwah,NJ: Lawrence Erlbaum Associates, Inc.
- Robins, A., Routree, J., & Rountree, N.(2003). Learning and teaching programming: A review and discussion. *Computer Science Education*,13(2), 137-172.
- Siti Soraya Abdul Rahman, & Benedict du Boulay. Learning programming via worked-examples. *In Proceedings of PPIG-WIP*, Dundee, 2010. Retrieved February 6, 2010, from <http://www.informatics.sussex.ac.uk/users/bend/cv/homepagenode6.html>
- Sweller, J., van Merriënboer, J.G., & Paas, F.G.W.C.(1998). Cognitive architecture and instructional design. *Educational Psychology Review*, 10 (3), 251-297.
- Tamara Van Gog., Fred Paas, b. & Jeroen J.G. Van Merriënboer.(2008). Effects of studying sequences of process-oriented and product-oriented worked examples on troubleshooting transfer efficiency. *Learning and Instruction*. 18(3), 211-222.
- Valente, J. A.(1995). Logo as a Window into the Mind. *Logo Update*, 4(1). Retrieved November 30, 2009, from <http://el.media.mit.edu/logo-foundation/pubs/logoupdate/>
- Van Gog, T., Paas, F., & Van Merriënboer, J. J. G.(2006). Effects of process-oriented worked examples on troubleshooting transfer performance. *Learning and Instruction*. 16, 154-164.
- Van Merriënboer, J. J. G., & Krammer, H. P. M. (1989). The completion strategy in programming instruction: Theoretical and empirical support. In S. Dijkstra, B. H. M. van Hout-Wolters, & P. C. van der Sijde(Eds.), *Research on Instruction* (pp. 45-61). Englewood wood Cliffs, NJ: Educational Technology Publications.
- Van Merriënboer, J. J. G. , & Paas, F.(1990).Automation and schema acquisition in learning

elementary computer programming. *Computers in Human Behavior*, (6), 273-289.

Van Merriënboer, J. J. G.(1990).Strategies for programming instruction in high school: Program completion vs. program generation. *Journal of Educational Communication Research*, 6(3), 265-

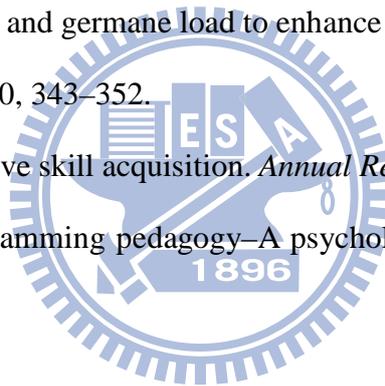
Van Merriënboer, J. J. G. & De Croock, M. B. M.(1992). Strategies for computer-based programming instruction: Program completion vs. program generation.*Journal of Educational Computing Research*, 8(3), 365-394.

Van Merriënboer, J. J. G., & Sweller, J.(2005). Cognitive load theory and complex learning: recent developments and future directions. *Educational Psychology Review*,17, 147-177.

Van Merriënboer, J. J. G., Kester, L., & Paas, F. (2006). Teaching complex rather than simple tasks:balancing intrinsic and germane load to enhance transfer of learning. *Applied Cognitive Psychology*, 20, 343–352.

VanLehn, K. (1996). Cognitive skill acquisition. *Annual Review of Psychology*, 47, 513-539.

Winslow, L. E.(1996). Programming pedagogy–A psychological overview. *SIGCSE Bulletin*, 28, 17-22.



附錄

附錄一：完成問題教學簡案

完成問題組以以設計電流急急棒遊戲為最終目的，教學流程即是完成問題策略的三步驟，分別為：追蹤觀察、模仿修改及擴充創作。以下分順序平行、重複執行以及條件執行三個子單元，表列出此次教學過程的教材簡案。

第一單元 順序平行

活動一 追蹤觀察（續下表）

教材分析	教材內容	
<p>順序執行概念</p> <p>說明：</p> <p>範例程式為一可執行的執行檔 (*.exe) 播放內容的順序依照圖 A、B、C、D、E 的順序出現。學生可以依照自己的速度，反覆觀看。</p> <p>範例程式說明：</p> <p>以會說規則的小恐龍為主角，學生觀察角色（小恐龍）的動作順序（小恐龍的動作如右圖所示）。</p>		<p>圖 A 小恐龍起始位置</p>
		<p>圖 B 小恐龍移動</p>
		<p>圖 C 小恐龍說話</p>
		<p>圖 D 小恐龍放大</p>
		<p>圖 E 小恐龍移縮小</p>
	<p>範 例</p>	<p>結 束</p>

學習單內容：觀察三張程式圖片，選出哪張圖最接近小恐龍的指令。

任務一：動動眼一點開「1 號小恐龍」 觀察角色(小恐龍)的順序

小恐龍的動作有：移動、放大、說話

哪一張圖最接近小恐龍的指令，把你覺得對的答案，在 內打勾，答案只有一個

<p>當 被點一下</p> <p>說 開始遊戲 3 秒</p> <p>在 1 秒內，滑行到 x: 17 y: 23</p> <p>將大小設定為 200 %</p> <p>等待 1 秒</p> <p>將大小設定為 100 %</p>	<p>當 被點一下</p> <p>在 1 秒內，滑行到 x: 17 y: 23</p> <p>說 開始遊戲 3 秒</p> <p>將大小設定為 200 %</p> <p>等待 1 秒</p> <p>將大小設定為 100 %</p>	<p>當 被點一下</p> <p>在 1 秒內，滑行到 x: 17 y: 23</p> <p>將大小設定為 200 %</p> <p>等待 1 秒</p> <p>將大小設定為 100 %</p> <p>說 開始遊戲 3 秒</p>
<p><input type="checkbox"/></p>	<p><input type="checkbox"/></p>	<p><input type="checkbox"/></p>

教材分析

教材內容

平行執行概念

說明：

範例程式為一可執行的執行檔播放內容的順序依照圖 A、B、C、D 的順序出現。學生可以依照自己的速度，反覆觀看。

範例程式說明：

觀察角色(小恐龍)同時做哪兩個動作(小恐龍的動作有：移動、放大、說話)。



圖 A
小恐龍放大



圖 B
小恐龍放大移動



圖 C
小恐龍縮小



圖 D
小恐龍說話

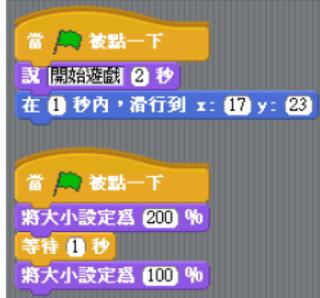
範 例 結 束

學習單內容：觀察三張程式圖片，選出哪張圖最接近小恐龍的指令。

任務二：動動眼一點開「2 號小恐龍」 觀察角色(小恐龍)的動作

小恐龍的動作有：移動、放大、說話

哪一張圖最接近小恐龍的指令，把你覺得對的答案，在 內打勾，答案只有一個

		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

活動二模仿修改

模仿修改活動中，教師給予學生的任務如表格所示，而任務的說明以註解的方式放置於完成問題程式的上方。

	順序執行	平行執行
教材分析	<p>完成問題內涵： 教師組合好「錯誤順序」的小恐龍。</p> <p>學生任務：依照任務，調整既有的程式順序。</p>	<p>完成問題內涵： 給予基本指令，但是缺少帽型控制指令 (hats)。</p> <p>學生任務：學生依照任務，把該一起執行的指令，組合好。</p>
教材內容	<p>****任務****</p> <p>希望小恐龍：</p> <ol style="list-style-type: none"> 1.先變大變小 2.移動到 x: 17 y:23的位置 3.說 開始遊戲 	<p>****完成任務****</p> <p>希望噴火龍：</p> <ul style="list-style-type: none"> 一邊 變大變小 一邊 移動到 x: 17 y:23的位置，說 開始遊戲 

活動三擴充創作

擴充創作活動中，教師給予學生的任務如表格所示，而任務的說明以註解的方式放置於完成問題程式的上方。

教材分析	教材內容
<p>順序執行</p> <p>擴充創作任務：</p> <p>小恐龍出現說完規則，小恐龍隱藏。</p> <p>(鼓勵學生按照自己的想法使用指令)</p> <p>教師給予的指令：</p> <p>[移動]滑行移動</p> <p>[外觀]隱藏、顯示、造型</p>	<p>任務：依照下面動作的順序 完成程式</p> <ol style="list-style-type: none"> 1.小恐龍 顯示 2.說：遊戲開始 3.滑行移動 4.說：點巫婆開始遊戲 5.小恐龍 隱藏 
<p>平行執行</p> <p>擴充創作任務</p> <p>在角色（小恐龍）的程式中，增加音效。讓自己在玩遊戲時，可以聽到聲音。</p> <p>教師給予的指令</p> <p>[聲音]播放聲音</p>	<p>教師事先準備好聲音，並告知學生如何使用聲音的指令。</p> 

第二單元 重複執行

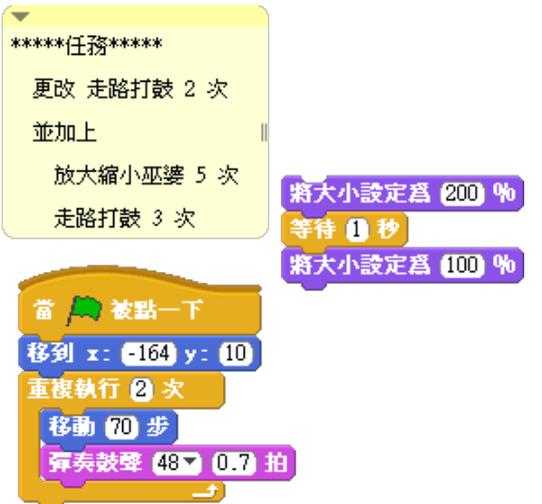
活動一 追蹤觀察（續下表）

教材分析	教材內容	
<p>重複執行 (repeat)</p> <p>說明： 範例程式圖 A 為範例程式的原始程式區塊。 圖 B 為範例程式舞台圖片。</p> <p>範例程式說明： 觀察角色 (巫婆) 重複幾次 (移動/打鼓) 的動作</p>		<p>圖 A 角色 (巫婆) 程式區塊</p>
		<p>圖 B 範例程式舞台</p>
範 例 結 束		
<p>學習單內容：根據指令重複的次數，選出正確的程式圖。</p>		
<p>任務(一) 動動眼：點開「重複動作」觀察角色 (巫婆)，回答下面問題</p>		
<div style="border: 1px solid black; padding: 10px;"> <p>問題一：右邊是 角色 (巫婆) 的動作→</p>  <p>想一想，巫婆總共重複圖中的動作幾次 下面哪一張圖，最接近巫婆的程式</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <input type="checkbox"/>  </div> <div style="text-align: center;"> <input type="checkbox"/>  </div> <div style="text-align: center;"> <input type="checkbox"/>  </div> </div> </div>		
教材分析	教材內容	
<p>重複執行 (forever)</p> <p>說明： 範例程式的另一角色 (十字架) 為學生觀察的重點，圖 A 為範例 程式的原始程式區塊。 圖 B 為範例程式舞台圖片。</p>		<p>圖 A 角色 (十字架) 程式區塊</p>

<p>範例程式說明： 觀察角色（十字架）的動作特性。</p>		<p>圖 B 範例程式舞台</p>
<p>範 例 結 束</p>		

<p>學習單內容：選出與角色（十字架）最接近的程式圖</p>
<p>問題二：哪個角色也是一直不停的動??觀察角色（十字架），回答下面問題</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">下面哪一張圖，最接近角色（十字架）的程式</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <input type="checkbox"/>  </div> <div style="text-align: center;"> <input type="checkbox"/>  </div> <div style="text-align: center;"> <input type="checkbox"/>  </div> </div> </div>

活動二模仿修改

	重複執行 (repeat)	重複執行 (forever)
<p>教材分析</p>	<p>完成問題內涵： 更改（移動/打鼓）指令的次數，加入指令（放大縮小）重複執行 2 次</p> <p>學生任務： 更改範例中的重複動作次數 增加不同的重複動作</p>	<p>完成問題內涵： 給予學生十字架.sprite，裡面已有旋轉指令</p> <p>學生任務： 匯入角色（十字架） 完成剛剛觀察到的角色（十字架）的程式。（學生可自行調整旋轉角度）</p>
<p>教材內容</p>		 <p style="text-align: center;">上圖為此任務的舞台圖</p>

活動三擴充創作

教材分析	教材內容
<p>重複執行 (repeat & forever)</p> <p>擴充創作任務：</p> <p>給予學生獨木橋.sprite</p> <p>鼓勵學生自己繪製障礙物，利用重複執行創作自己的障礙物。</p> <p>教師給予的指令：</p> <p>[外觀]隱藏 / 顯示</p> <p>[控制]等待 1 秒</p> <p>[移動]滑行移動、旋轉</p>	<p>獨木橋(顯示/隱藏)重複10次， 最後獨木橋是要可以出現，讓巫婆過去的。 想辦法加入重複執行，讓指令更簡短。</p> 

第三單元 條件執行：

活動一追蹤觀察 (續下表)

<p>開始畫面：播放順序由圖 A 到圖 B，以下說明範例使用的方法。</p>	
 <p>圖 A</p>	 <p>圖 B</p>
教材分析	教材內容
<p>條件執行 (if)</p> <p>說明：</p> <p>當學生點角色 (巫婆)，角色會移動到滑鼠上，學生必須像玩電流急急棒遊戲一樣，點主角巫婆，碰觸其它角色，觀察角色的動作。</p> <p>範例程式說明：</p> <p>圖 C 為角色 (巫婆) 碰到角色 (十字架) 的情況：碰到角色 (十字架) 角色 (巫婆) 說：扣 1 分。</p>	 <p>圖 C</p> <p>範 例 結 束</p>

學習單內容：巫婆 碰到十字架 就 說扣 1 分。根據說明以及觀察到的情形，選出正確的程式圖。

問題二：如果....就.....

下面哪一張圖，最接近角色(巫婆)的程式，在□內打勾
狀況：巫婆 碰到十字架 就 說扣1分

<input type="checkbox"/> 	<input type="checkbox"/> 	<input type="checkbox"/> 
--	--	--

教材分析

教材內容

條件執行 (if-then)

範例程式說明：

碰到角色(紅妖怪)就放大
沒有碰到角色(紅妖怪)回復正常



圖 A
角色(巫婆)
未碰到角色
(紅妖怪)



圖 B
角色(巫婆)
碰到角色(紅
妖怪)

範 例 結 束

學習單內容：巫婆碰到紅妖怪 就 變大 否則 變回原狀，根據說明以及觀察到的情形，選出正確的程式圖。

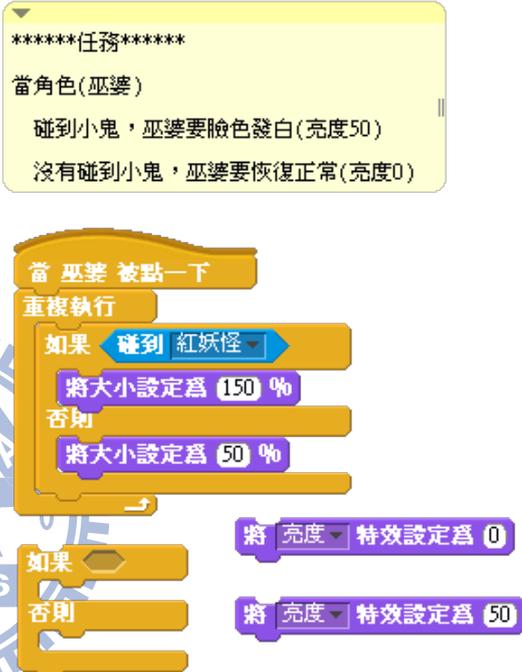
任務(一)動動眼：點開桌面上的「巫婆奇遇記」，拿起角色(巫婆)碰碰看其他角色

問題一：如果...就....否則...

下面哪一張圖，最接近角色(巫婆)的程式，在□內打勾
狀況：巫婆碰到紅妖怪 就 變大 否則 變回原狀

<input type="checkbox"/> 	<input type="checkbox"/> 	<input type="checkbox"/> 
--	--	--

活動二模仿修改

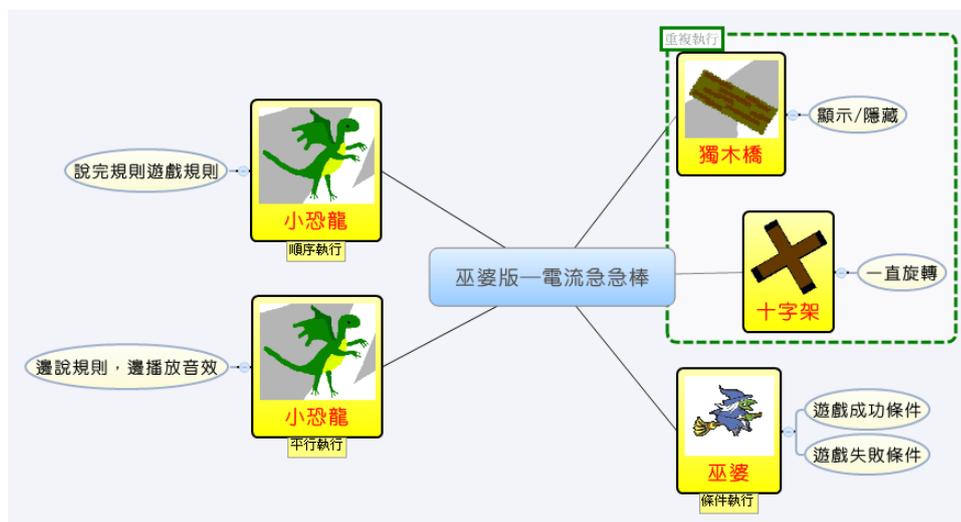
	條件執行 (if)	條件執行 (if-then)
教材分析	<p>完成問題內涵： 給予角色動作，學生根據任務加入偵測的條件。</p> <p>學生任務： 當角色（巫婆）碰到帽子，巫婆要說：過關了。</p>	<p>完成問題內涵： 給予角色動作以及 if 基本框架，學生根據任務加入偵測的條件。</p> <p>學生任務： 當角色（巫婆）碰到小鬼，巫婆要臉色發白，沒有碰到小鬼，巫婆要恢復正常。</p>
教材內容	 <p>上圖為條件任務的舞台圖</p> 	

活動三擴充創作

教材分析	教材內容
<p>條件執行 (if & if-then)</p> <p>擴充創作任務： 給予學生巫婆.sprite 鼓勵學生自己更換角色造型，利用條件執行，創造自己的遊戲規則。</p> <p>教師給予的指令： 全部停止（遊戲過關/失敗的條件） [偵測]當碰到顏色■ （利用顏色來設定遊戲規則）</p>	

附錄二：問題導向教學簡案

問題導向組的學習教材以設計電流急急棒遊戲為最終目的，教學流程即是問題解決的步驟，分別為：瞭解問題、分析問題、撰寫程式、運作與觀察程式。下表為此次教學的程式概念圖。



教學簡案：

週次	教學重點	教學內容	備註
第四週	順序執行	<p>說明：指令的執行是有順序的</p> <p>了解問題：(揭示任務)還記得會說規則的小恐龍，寫一個程式讓小恐龍依照下面的指令來做動作，順序不能有錯。</p> <p>小恐龍，說：「你好」</p> <p>接著播放聲音(ComputerBeeps)</p> <p>小恐龍放大縮小，</p> <p>小恐龍，說「點巫婆開始遊戲」。</p> <p>分析問題：(全班討論)</p> <ul style="list-style-type: none"> ※ 問一：請問這些步驟利用到哪些「程式盒子」。 ※ 問二：分別是哪些指令。(說話、撥聲音、放大縮小) ※ 問三：會播放聲音嗎？ ※ 問四：會放大縮小嗎？ <p>實際操作：請學生點開小恐龍.sb</p> <p>學生完成任務</p> <p>運作與觀察程式：你有達成任務嗎？</p>	
第五週	順序執行	<p>說明：指令的執行是有順序的</p> <p>了解問題：播放範例。</p> <p>顯示</p> <p>播放音效</p> <p>小恐龍要說的規則：</p> <p>請幫助巫婆拿到巫婆帽</p> <p>放大縮小</p> <p>但是不能碰到牆壁</p> <p>開始遊戲</p>	

		<p>隱藏</p> <p>給學生觀看範例(如上)小恐龍.exe</p> <p>學生任務：依照看到的例子，排列出你的小恐龍。</p> <p>※ 範例程式中的小恐龍有哪些動作。(出現、說話、隱藏)</p> <p>分析問題：(全班討論)</p> <p>※ 問一：請問這些步驟利用到哪些「程式盒子」。</p> <p>※ 問二：分別是哪些指令。(說話、撥聲音、放大縮小)</p> <p>※ 問三：會播放聲音嗎？</p> <p>※ 問四：會放大縮小嗎？</p> <p>※ 問一：請問這些步驟利用到哪些「程式盒子」。</p> <p>※ 問二：分別是哪些指令。(說話、顯示隱藏)</p> <p>※ 問三：如果小恐龍要一開始出現，之後消失，那程式要怎麼排列。(關鍵概念：順序執行)</p> <p>實際操作：請學生點開小恐龍.sb(規則已經打好了)</p> <p>學生完成任務</p> <p>運作與觀察程式：你有達成任務嗎？</p> <p>測驗</p>	
第六週	重複執行	<p>說明：重複執行的指令是用在某些動作要執行很多次的時候使用。</p> <p>了解問題：</p> <p>巫婆移動 70 步，打鼓 1 次，這個動作重複 6 次。</p> <p>想辦法加入重複執行，讓指令更簡短。</p> <p>分析問題：(全班討論)</p> <p>※ 問一：請問這些步驟利用到哪些「程式盒子」。</p> <p>※ 問二：分別是哪些指令。(移動/打鼓)</p> <p>※ 問三：任務的程式重複幾次？</p> <p>※ 問四：怎樣加入重複執行的指令。(關鍵概念：重複執行)</p> <p>實際操作：請學生點開巫婆.sb</p> <p>學生完成任務</p> <p>運作與觀察程式：你有達成任務嗎？</p>	
第七週	重複執行	<p>了解問題：還記得遊戲中的獨木橋嗎？</p> <p>獨木橋(顯示/隱藏)重複 10 次，</p> <p>最後獨木橋是要可以出現，讓巫婆過去的。</p> <p>想辦法加入重複執行，讓指令更簡短。</p> <p>分析問題：(全班討論)</p> <p>※ 問一：請問這些步驟利用到哪些「程式盒子」。</p> <p>※ 問二：分別是哪些指令。(隱藏/顯示)</p> <p>※ 問三：顯示/隱藏要怎麼排序，才能讓獨木橋最後是出現的。(隱藏/顯示、加上等待秒數)</p> <p>※ 問四：怎樣加入重複執行的指令。(關鍵概念：重複執行)</p> <p>實際操作：請學生點開獨木橋.sb</p>	

		<p>學生完成任務</p> <p>運作與觀察程式：你有達成任務嗎？</p> <p>測驗</p>	
第八週	條件執行	<p>說明：如果(條件).....(不滿足條件)就....</p> <p>了解問題：觀察範例</p> <p>遊戲中的巫婆，碰到紅妖怪就會放大；沒有碰到紅妖怪就恢復正常大小。</p> <p>遊戲中的巫婆，碰到小鬼就會臉色發白；沒有碰到鬼就恢復正常。</p> <p>※ 問一：巫婆碰到紅妖怪會發生什麼事？</p> <p>※ 問二：巫婆碰到小鬼會發生什麼事？</p> <p>分析問題：(全班討論)</p> <p>※ 問一：請問這些步驟利用到哪些「程式盒子」。</p> <p>※ 問二：分別是哪些指令。(大小、鬼的特效)</p> <p>※ 問三：怎樣加入(如果.否則)指令。(關鍵概念：條件執行)</p> <p>■ 碰到紅妖怪...[動作]；沒有碰到紅妖怪...[動作]</p> <p>■ 碰到小鬼...[動作]；沒有碰到小鬼...[動作]</p> <p>※ 問四：條件判斷要加上「重複執行」→要一直檢驗條件有沒有成立。</p> <p>※ 問五：放大之後要如何恢復原來的大小。</p> <p>實際操作：請學生點開巫婆.sb</p> <p>學生完成任務</p> <p>運作與觀察程式：你有達成任務嗎？</p>	
第九週	條件執行	<p>說明：如果(條件)...</p> <p>了解問題：觀察範例</p> <p>遊戲中的巫婆，怎樣才算過關；什麼條件下才會失敗。</p> <p>※ 問一：怎樣才算過關？</p> <p>※ 問二：怎樣才算失敗？</p> <p>分析問題：(全班討論)</p> <p>※ 問一：請問這些步驟利用到哪些「程式盒子」。</p> <p>※ 問二：分別是哪些指令。(偵測指令與條件指令)</p> <p>※ 問三：怎樣加入(如果...)指令。(關鍵概念：條件執行)</p> <p>※ 問四：條件判斷要加上「重複執行」→要一直檢驗條件有沒有成立。</p> <p>實際操作：請學生點開巫婆.sb</p> <p>運作與觀察程式：你有達成任務嗎？</p>	

第十週：進行電流急急棒遊戲組合。

附錄三：基本指令教學

第一單元 認識 Scratch

班級：四年__班 座號：__ 姓名：__

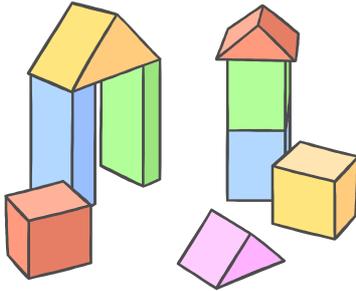
Scratch (貓爪) 是一套新的程式語言，讓寫程式就像玩積木一樣簡單。你可以用它來創造互動式故事、動畫、遊戲。

一、 開始認識貓吧！



- ◇ 程式盒子：切換不同功用的指令(動作、外觀、聲音..)
- ◇ 程式區塊：各種指令
- ◇ 角色資訊：角色的名字、位置、方向、允許翻轉..
- ◇ 程式組合區：組合指令
- ◇ 舞台：角色表演的地方
- ◇ 角色清單：新增角色、開啟角色、列出有哪些角色

寫程式……就像……「玩積木」一樣

選好積木(程式區塊)	發揮創意(組合程式區)	遊戲、故事、動畫(舞台)
		

二、程式盒子：各種功用的指令—動作、外觀、聲音、控制、偵測

動作	外觀	聲音
控制角色的動作	控制角色的外觀變化	特殊聲音、鼓聲、鋼琴聲

三、新增角色 與 匯入角色造型

新增角色

加入新的演員，Scratch 中有很多演員可以選擇。

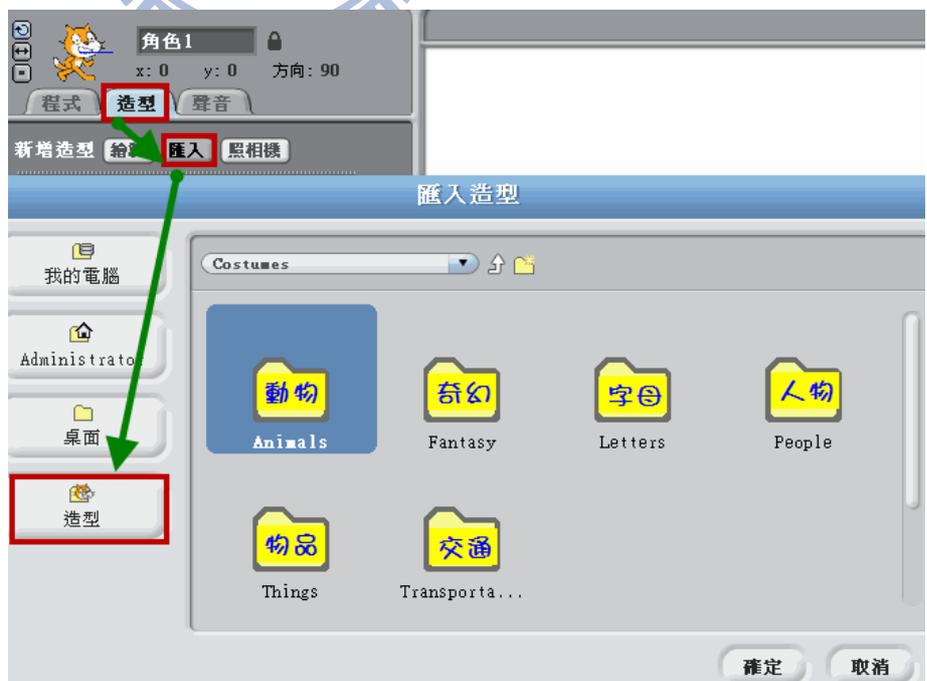
演員清單



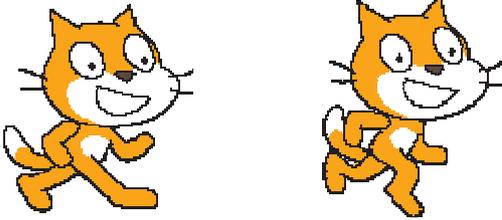
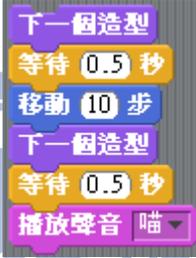
匯入角色造型

針對角色，加入新的造型，Scratch 中有各類造型可以選擇。

角色造型



四、貓刀小試：點開那隻「貓」，試試看，你可以完成下面任務嗎

任務區	程式組合區
<p>主題：小貓走路</p>  <p>造型 1 造型 2</p> <p>※造型的切換，中間要加上等待秒數</p>	<p>按照下方程式組合，點兩下執行。</p>  <p>讓小貓繼續在原地換造型</p> 
<p>準備好了嗎？</p>	<p>按照下方程式組合，點兩下執行。</p>  <p>小貓看起來會像是 「在走路」</p> <p><input type="checkbox"/> 你完成了嗎？</p>

五、寫寫看：

按照右圖回答下面問題：
(填代號 A、B……)

- ☺ () 角色表演的地方。
- ☺ ()  指令，要到圖中的哪個地方找。
- ☺ () Scratch 中拼湊指令、創作程式的地方。
- ☺ () 切換不同功用的指令
- ☺ () 新增角色。



六、跟 Scratch 的第一次接觸，留下你的想法，讓老師知道

我覺得 Scratch 是個好玩的軟體，我很期待後面的課程。……

非常	符	普	不	非常
	符		符	不
符合	合	通	合	符合
<input type="checkbox"/>				

附錄四：各單元形成性測驗

第一單元 同時執行 與 順序執行

你知道嗎……

按照自己想法，**清楚、正確**的對角色下指令，在程式組合區中**按照順序**的排列指令，角色就會依照自己的想法動起來！

注意!!

下面總共有 5 題 **選擇題**。

依照給的圖片，如：，回答圖片後面的問題。

班級：四年____班 座號：_____ 姓名：_____

測 驗 開 始



() 1、當貓被點一下，角色（貓）的動作順序會是？

- (1) 喵一聲接著說：「你好」 (2) 說：「你好」接著喵一聲
(3) 沒反應 (4) 喵一聲。



() 2、當貓被點一下，角色（貓）的動作會是？

- (1) 喵一聲接著說：「你好」 (2) 說：「你好」接著喵一聲
(3) 沒反應 (4) 邊喵一聲，邊說：「你好」。



() 3、角色（蜜蜂）原本的大小 →



當蜜蜂被點一下，角色（蜜蜂）的動作會是？

- (1) 變大 (2) 變小
(3) 先變大再變小 (4) 不會變。

~ ~ 背 面 還 有 題 目 加 油 ! ~ ~

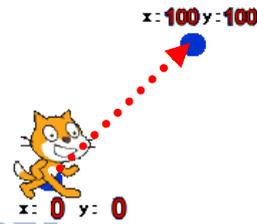
```

當 被點一下
顯示
說 你好! 2 秒
隱藏

```

4、右邊是角色（小恐龍）的程式，回答下面問題：

- () 當綠旗被點一下時，小恐龍一開始在舞台上的動作是？
 - (1) 沒有反應 (2) 出現 (3) 隱藏 (4) 說：「你好」。
- () 執行圖片中的指令時，小恐龍的動作的順序會是？
 - (1) 小恐龍不見，說：「你好」，再消失不見 (2) 從頭到尾看不見角色
 - (3) 小恐龍出現，說：「你好」，小恐龍不見 (4) 小恐龍出現，小恐龍不見，說：「你好」。



() 5、右圖是 角色（貓）在舞台上的位置，x: 0 y: 0 x: 100 y: 100 （箭頭是貓移動的方向）

如果希望角色（貓），在同一時間一邊移動、一邊說話

下面哪段指令可以做到。

(1)

```

當 被點一下
在 1 秒內，滑行到 x: 0 y: 0
說 你好! 2 秒
在 1 秒內，滑行到 x: 100 y: 100

```

(2)

```

當 被點一下
在 1 秒內，滑行到 x: 0 y: 0
在 1 秒內，滑行到 x: 100 y: 100
當 被點一下
說 你好! 2 秒

```

(3)

```

當 被點一下
在 1 秒內，滑行到 x: 0 y: 0
在 1 秒內，滑行到 x: 100 y: 100
說 你好! 2 秒

```

(4)

```

當 被點一下
在 1 秒內，滑行到 x: 0 y: 0
在 1 秒內，滑行到 x: 100 y: 100
當 被點一下
說 你好! 2 秒

```

測 驗 結 束

很棒喔！記得要寫上姓名，最後不要忘記交卷！

第二單元 重複動作

善用重複執行的指令，可以利用很短的「程式」一次讓角色「重複執行」很多次你想要的動作。



下面總共有 5 題 **選擇題**。依照給的圖片，回答圖片後面的問題。

班級：四年____班 座號：_____ 姓名：_____

測 驗 開 始

1、依照圖片，回答下面題目：

- () 1.1 當綠旗被點一下，角色的動作會是？
- (1) 連續 2 次「移動 10 步，打鼓 1 拍，說你好」
 - (2) 連續 3 次「移動 10 步，打鼓 1 拍，說你好」
 - (3) 連續 2 次「移動 10 步，打鼓 1 拍」，接著說你好
 - (4) 連續 3 次「移動 10 步，打鼓 1 拍」，接著說你好

() 1.2 下面哪張圖片執行的結果，跟上面是一樣的

(1)

(2)

(3)

(4)

背 面 還 有 題 目 加 油 ！



() 2、左邊的指令代表的是？

- (1) 每 0.2 拍，彈奏鼓聲 10 次 (2) 重複執行 10 次彈奏鼓聲的指令
 (3) 重複執行 48 次彈奏鼓聲的指令 (4) 每 0.2 拍，彈奏鼓聲 1 次



() 3、當綠旗被點一下，角色(蝙蝠)的動作會是

- (1) 原地走路 4 次 (2) 揮動翅膀往前飛 4 次
 (3) 原地揮動翅膀 4 次 (4) 原地揮動翅膀不停止。



() 4、左邊圖片跟第三題比較，角色(蝙蝠)的動作

有什麼不同？

- (1) 沒有不同 (2) 變成 揮動翅膀往前飛 5 次
 (3) 變成 原地揮動翅膀 5 次 (4) 變成 原地揮動翅膀不會停



() 5、左邊的圖代表

- (1) 角色 旋轉 5 次 (2) 角色 不停的移動
 (3) 角色 移動 5 步 (4) 角色 不停的旋轉

測 驗 結 束

很棒喔！記得寫上姓名，不要忘記交卷！

第三單元 條件執行

利用 ，讓你的角色根據設定好的條件，執行動作。

題目都是「**選擇題**」**答案只有一個**。依照給的圖片，回答圖片後面的問題。

班級：四年____班 座號：_____ 姓名：_____

測 驗 開 始

() 1、  左邊圖中  的區域應該擺放的是？

(1) 角色要執行的動作 (2) 角色要偵測的條件 (3) 等待的秒數 (4) 執行的次數。

() 2、想要完成『如果 角色(巫婆)碰到紅妖怪 就變大，否則 恢復原狀』這段程式，該使用下列哪個關鍵的指令？

(1)  (2)  (3)  (4) 

3、  依照圖片，回答下面題目：

() 3.1 以上這些指令是放在哪個角色裡面？(1) 小鬼 (2) 巫婆 (3) 紅妖怪 (4) 舞台。

() 3.2 在哪一種「偵測條件」下，角色(巫婆)會變大？

(1)  (2)  (3)  (4) 

() 3.3 什麼條件下，角色(巫婆)的大小會「恢復正常」(也就是巫婆大小 100)？

(1) 碰到小鬼 (2) 沒有碰到小鬼 (3) 碰到紅妖怪 (4) 沒有碰到紅妖怪。

背 面 還 有 題 目 加 油 !

() 4、想要完成『如果 角色(巫婆) 碰到紅妖怪就變小』這段程式，

該使用下列哪個關鍵的指令？

- (1)  (2)  (3)  (4) 



() 5、左邊是角色(十字架) 的程式，執行這段指令代表的是？

- (1) 如果巫婆碰到十字架，巫婆說扣 1 分 (2) 十字架說碰到巫婆
 (3) 如果巫婆碰到十字架，十字架說扣 1 分 (4) 什麼事都不會發生。



6、

上面是角色(巫婆) 在「巫婆版—電流急急棒」中的一段程式，回答下面問題：

() 6.1 這段程式代表的是

- (1) 遊戲過關的條件 (2) 遊戲失敗的條件 (3) 被扣 1 分 (4) 遊戲正要開始。

() 6.2 圖片中的  指令功用是？

- (1) 不斷的偵測「條件  」有沒有發生 (2) 讓程式不要執行太快
 (3) 讓程式執行慢一點 (4) 設定判斷條件用的。

測 驗 結 束

很棒喔！記得寫上姓名，不要忘記交卷！

附錄五：總結性測驗

恭喜你(妳)，通過了貓爪的考驗，你已經學到了基本的「程式設計」。

下面的題目是測試自己有沒有學會「正確的對電腦下指令」，請用程式設計師的想法，回答下面的問題。

測 驗 開 始

() 1、要完成「當按下綠旗，貓出現，說你好後就消失」。下面哪張圖是正確的指令？

(1)  (2)  (3)  (4) 

() 2、當綠旗被點一下，角色的動作是？

- (1) 滑行移動，說遊戲開始，放大縮小 (2) 滑行移動，放大縮小，說遊戲開始
 (3) 滑行移動，說遊戲開始，縮小放大 (4) 滑行移動，縮小放大，說遊戲開始

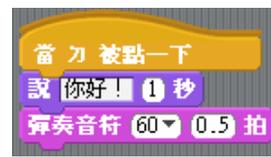


() 3、當貓被點一下，角色（貓）的動作是？

- (1) 連續 5 次「移動、打鼓」，接著說你好 (2) 連續 5 次「移動、打鼓、說你好」
 (3) 連續 5 次「打鼓、移動」，接著說你好 (4) 連續 5 次「打鼓、移動、說你好」。

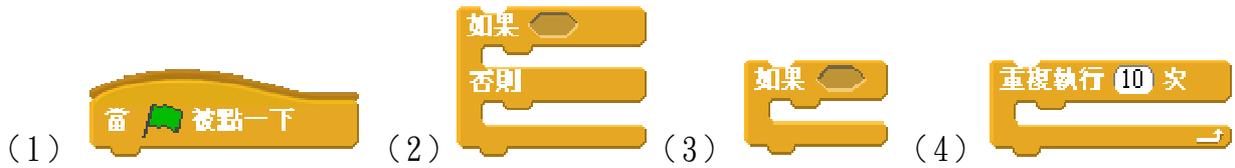


() 4、下面哪張圖是「當角色(ㄉ)被點到，可以邊彈奏音符 Do，邊說話」？

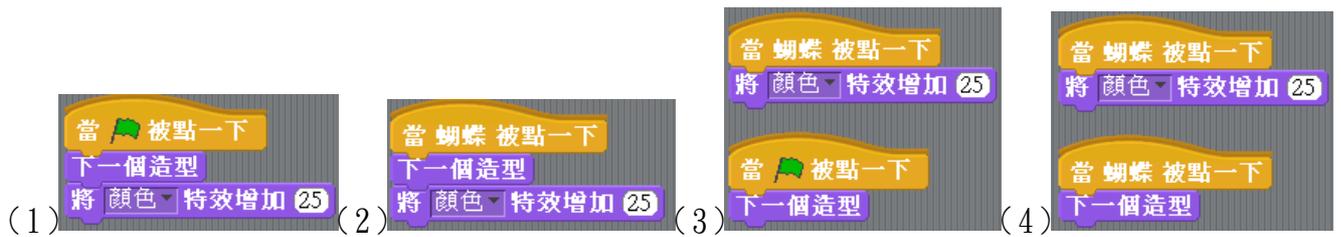
(1)  (2)  (3)  (4) 

() 5、完成「角色(狗) 碰到 角色(貓) 會變大，沒有碰到 會變回原狀」

這段程式，會利用到哪個指令。



() 6、下面哪張圖是「當角色(蝴蝶)被點一下，蝴蝶一邊換造型，一邊變顏色」？



() 7、當蝙蝠被點一下，角色(蝙蝠)的動作看起來會像？

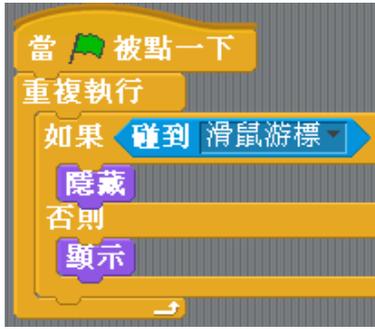


- (1) 蝙蝠先揮動翅膀，之後再滑行移動 (2) 一邊揮動翅膀，一邊滑行移動
(3) 只會揮動翅膀 (4) 只會滑行移動

() 8、當綠旗被點一下時，角色的動作會是？

- (1) 不停的旋轉 (2) 先說你好，接著不停的旋轉
(3) 不停的說你好以及旋轉 (4) 連續旋轉 5 次。





() 9、依照圖中的程式找出下列哪個選項是**錯誤**的？

- (1) 圖中的判斷條件是「角色是否碰到滑鼠游標」
- (2) 當滑鼠游標碰到角色，角色會不見
- (3) 當綠旗被點一下，角色會不見
- (4) 當滑鼠游標沒有碰到角色，角色會出現

() 10、下面哪個選項的敘述，會用到像：  的偵測指令？

- (1) 角色不停的旋轉
- (2) 角色重複彈奏鼓聲 5 次之後，說你好
- (3) 如果角色碰到滑鼠游標，會不見
- (4) 角色黏在滑鼠游標上



() 11、請問角色(忍者) **最後的動作**是？

- (1) 忍者出現，說你好
- (2) 忍者說你好後，隱藏
- (3) 忍者隱藏，看不到忍者說話
- (4) 忍者出現，不說話。

```

當 被點一下
移動 10 步
彈奏鼓聲 48 1 拍
說 你好！ 1 秒
移動 10 步
彈奏鼓聲 48 1 拍
說 你好！ 1 秒
移動 10 步
彈奏鼓聲 48 1 拍
說 你好！ 1 秒

```

() 12、利用「重複執行」縮短左邊的指令，下面哪張圖是正確的？

(1)

```

當 被點一下
重複執行 3 次
移動 10 步
彈奏鼓聲 48 1 拍
說 你好！ 1 秒

```

(2)

```

當 被點一下
移動 10 步
彈奏鼓聲 48 1 拍
說 你好！ 1 秒

```

(3)

```

當 被點一下
重複執行
移動 10 步
彈奏鼓聲 48 1 拍
說 你好！ 1 秒

```

(4)

```

當 被點一下
重複執行 3 次
移動 10 步
彈奏鼓聲 48 1 拍
說 你好！ 1 秒

```

() 13、下面哪個敘述，要**一定要**用到指令  才可以完成？

- (1) 如果角色碰到滑鼠游標，會消失；滑鼠移開，角色就出現
- (2) 如果角色碰到滑鼠游標，會不見
- (3) 點角色一下，角色開始移動
- (4) 角色不停的旋轉

() 14、下面哪個選項的程式，會使用到  指令

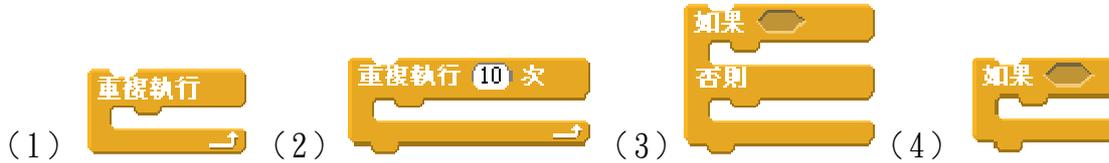
- (1) 角色不停的旋轉
- (2) 角色重複彈奏鼓聲 5 次之後，說你好
- (3) 如果角色碰到滑鼠游標，會不見
- (4) 角色黏在滑鼠游標上

現在你已經是電流急急棒的遊戲設計者了

請用程式設計者的想法回答 15、16、17 題

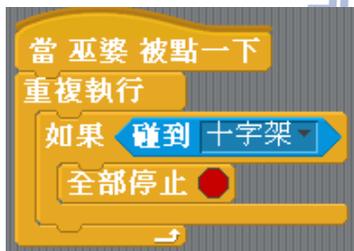
() 15、遊戲中有很多的障礙物都需要不停的移動，或是旋轉。

這是利用到下面哪個指令？



() 16、電流急急棒的規則是主角碰到「黑色的牆壁」，遊戲會結束。

下面哪個是正確的判斷條件？



() 17、左圖為角色(巫婆)的程式。圖中的程式代表？

- (1) 如果點到巫婆，遊戲停止
- (2) 如果十字架碰到巫婆，遊戲停止
- (3) 如果巫婆碰到十字架，遊戲停止
- (4) 角色(巫婆)一直重複「碰到十字架」。

圖中是角色(巫婆) 在「巫婆版—電流急急棒」中的一段程式



依照圖片回答 18、19、20 題

() 18、圖中的程式代表的是？

- (1) 遊戲過關的條件
- (2) 遊戲失敗的條件
- (3) 被扣 1 分
- (4) 遊戲正要開始。

() 19、當巫婆碰到巫婆帽時，會有什麼情況發生？選出**正確**的選項。

- (1) 巫婆帽說拿到了
- (2) 程式全部停止，巫婆說拿到了
- (3) 巫婆帽說全部停止
- (4) 巫婆說拿到了，程式全部停止

() 20、圖中的  指令功用是？

- (1) 不斷的偵測「條件  」有沒有發生
- (2) 讓程式不要執行太快
- (3) 讓程式執行慢一點
- (4) 設定條件判斷用的。

附錄六：學生學習態度問卷(問題導向組)

「Scratch 基本程式設計」課程問卷

在經過了幾堂「Scratch 基本程式設計」的課程之後，老師想要了解大家對於這次課程的想法。請仔細讀過每一題之後，勾選出最符合自己想法的選項

班級：401 座號：_____ 姓名：_____ ※性別：男 女

一、基本態度：

※我對「電腦」的感覺是 非常喜歡 喜歡 不喜歡 非常不喜歡

※我對學校的電腦課的感覺是 非常喜歡 喜歡 不喜歡 非常不喜歡

二、關於這次課程，你的想法是……

題目	4	3	2	1
	非常同意	同意	不同意	很不同意
與應用軟體（如 Word、PowerPoint）相比，我覺得程式設計課程比較有趣				
我覺得學習程式設計 很無趣				
上程式設計課時，我覺得很開心				
我很期待下一次課程的到來				
經過此次課程，引起我對程式設計的興趣				
我以後還會繼續使用 Scratch 來創作作品				
透過老師分析角色的動作，讓我了解到基本程式設計的概念				
透過老師上課前，一步步分析要怎麼完成任務，讓我更容易完成我的遊戲				
透過這次課程，我了解到基本的程式設計概念				
我覺得完成「會說規則的小恐龍」的困難度是 <input type="checkbox"/> 很容易 <input type="checkbox"/> 容易 <input type="checkbox"/> 有點難 <input type="checkbox"/> 非常困難				
我覺得完成「旋轉十字架以及獨木橋」的困難度是 <input type="checkbox"/> 很容易 <input type="checkbox"/> 容易 <input type="checkbox"/> 有點難 <input type="checkbox"/> 非常困難				
我覺得完成「電流急急棒遊戲規則」的困難度是 <input type="checkbox"/> 很容易 <input type="checkbox"/> 容易 <input type="checkbox"/> 有點難 <input type="checkbox"/> 非常困難				

最後請你用一句話寫下，你對於「學習程式設計」的感想

附錄七：學生學習態度問卷(一般教學組)

「Scratch 基本程式設計」課程問卷

在經過了幾堂「Scratch 基本程式設計」的課程之後，老師想要了解大家對於這次課程的想法。請仔細讀過每一題之後，勾選出最符合自己想法的選項

班級：402 座號：_____ 姓名：_____ ※性別：男 女

一、基本態度：

※我對「電腦」的感覺是 非常喜歡 喜歡 不喜歡 非常不喜歡

※我對學校的電腦課的感覺是 非常喜歡 喜歡 不喜歡 非常不喜歡

二、關於這次課程，你的想法是……

題目	4	3	2	1
	非常同意	同意	不同意	很不同意
與應用軟體（如 Word、PowerPoint）相比，我覺得程式設計課程比較有趣				
我覺得學習程式設計很無趣				
上程式設計課時，我覺得很開心				
我很期待下一次課程的到來				
經過此次課程，引起我對程式設計的興趣				
我以後還會繼續使用 Scratch 來創作作品				
透過老師講解每個指令的用法，讓我了解到基本程式設計的概念				
透過老師上課前，講解指令的用法，讓我更容易完成我的遊戲				
透過這次課程，我了解到基本的程式設計概念				
我覺得完成「會說規則的小恐龍」的困難度是 <input type="checkbox"/> 很容易 <input type="checkbox"/> 容易 <input type="checkbox"/> 有點難 <input type="checkbox"/> 非常困難				
我覺得完成「旋轉十字架以及獨木橋」的困難度是 <input type="checkbox"/> 很容易 <input type="checkbox"/> 容易 <input type="checkbox"/> 有點難 <input type="checkbox"/> 非常困難				
我覺得完成「電流急急棒遊戲規則」的困難度是 <input type="checkbox"/> 很容易 <input type="checkbox"/> 容易 <input type="checkbox"/> 有點難 <input type="checkbox"/> 非常困難				

最後請你用一句話寫下，你對於「學習程式設計」的感想

附錄八：學生學習態度問卷(完成問題組)

「Scratch 基本程式設計」課程問卷

在經過了幾堂「Scratch 基本程式設計」的課程之後，老師想要了解大家對於這次課程的想法。請仔細讀過每一題之後，勾選出最符合自己想法的選項

班級：403 座號：_____ 姓名：_____ ※性別：男 女

一、基本態度：

※我對「電腦」的感覺是 非常喜歡 喜歡 不喜歡 非常不喜歡

※我對學校的電腦課的感覺是 非常喜歡 喜歡 不喜歡 非常不喜歡

二、關於這次課程，你的想法是……

題目	4	3	2	1
	非常同意	同意	不同意	很不同意
與應用軟體（如 Word、PowerPoint）相比，我覺得程式設計課程比較有趣				
我覺得學習程式設計很無趣				
上程式設計課時，我覺得很開心				
我很期待下一次課程的到來				
經過此次課程，引起我對程式設計的興趣				
我以後還會繼續使用 Scratch 來創作作品				
透過每堂課會先看「範例」，並完成學習單，讓我了解到基本程式設計的概念				
透過第一堂課的「例子」，以及完成老師給我的「部分任務」，讓我更容易完成我的遊戲				
透過這次課程，我了解到基本的程式設計概念				
我覺得完成「會說規則的小恐龍」的困難度是 <input type="checkbox"/> 很容易 <input type="checkbox"/> 容易 <input type="checkbox"/> 有點難 <input type="checkbox"/> 非常困難				
我覺得完成「旋轉十字架以及獨木橋」的困難度是 <input type="checkbox"/> 很容易 <input type="checkbox"/> 容易 <input type="checkbox"/> 有點難 <input type="checkbox"/> 非常困難				
我覺得完成「電流急急棒遊戲規則」的困難度是 <input type="checkbox"/> 很容易 <input type="checkbox"/> 容易 <input type="checkbox"/> 有點難 <input type="checkbox"/> 非常困難				

最後請你用一句話寫下，你對於「學習程式設計」的感想