

國立臺灣師範大學資訊教育研究所
碩士論文

指導教授：吳正己 博士

程式設計對初學者問題解決能力之影響
—以機器車程式設計為例

研究生：鮑志軒 撰

中華民國一百零二年七月

摘要

本研究探討機器車輔助程式設計學習對初學者問題解決能力及學習態度的影響。研究採準實驗設計，實驗時間共十週，參與者為南部某國中國一學生共二班 63 人，其中一班學生為實驗組，進行樂高機器車程式設計學習活動；另一班為控制組，進行傳統計算機概論教學活動。資料蒐集及分析包括「新編問題解決能力測驗」得分，以及態度問卷上封閉及開放式問題之填答。

研究結果發現：（1）機器車輔助程式設計對學生之問題解決能力無顯著影響；（2）女性學生對機器車程式設計較無興趣；（3）男性學生對機器車程式設計有較高的興趣，在實驗後，其「解決方法」及「有效性」二個問題解決面向之能力有顯著提升。男性學生經過機器車程式設計學習後，能提出更多樣及有效的問題解決構想，並且思考變得更周密，也更善於運用既有的知識及經驗來掌握關鍵要素及有效的方法。建議未來可重覆本研究，但增加與其它程式設計情境之比較，例如與 Scratch、Alice 等。此外，亦可運用質性方法深入探討程式設計與問題解決能力各面向之因果關係。

關鍵字：樂高機器車、程式設計、問題解決。

Abstract

The Effects of Robot Programming in Promoting Students` Problem Solving Ability

By

Chih-Hsuan Pao

This study investigated the effects of using robots to assist students learning programming, in particular, how robot programming might affect students` problem solving ability. A quasi-experimental design was implemented in this study. Two classes of 63 7th grade students from a middle school in south Taiwan participated in this study. One class of the students served as the experiment group, who had robot programming activities; the other class served as the control group, who had traditional learning activities on introduction of computer science throughout the experiment. The experiment lasted 10 weeks, with 2 hours per week, totaling 20 hours.

The findings of this study were: (1)The robot programming approach had no effects on promoting students` problem solving ability. (2)Female students were not interested in robot programming. (3)Male students showed interests in robot programming and their problem solving abilities were enhanced in both solution-finding and solution-effectively dimensions. We recommended future research repeated this study with a longer duration of experiment and with qualitative methods to investigate the casual-effect relationship of robot programming and problem solving.

Keywords: Problem-solving, Programming, Robot

誌謝

本文得以順利付梓，首先要感謝指導教授 吳正己 博士，在研究過程中不遺餘力的悉心指導，讓我在學術研究上獲益良多。由於我本身並不是教育背景出生，因此在學習的道路上總是坎坎坷坷，還好有吳老師不斷的在一旁給予我意見，教導我論文的正確寫法，我才能在今天把論文完善。

其次，我要感謝我的家人和研究室裡的夥伴，我的家人在這幾年來給予我不遺餘力的支持，並且一直勉勵我不要放棄，繼續堅持下去，一定能順利畢業；而研究室的夥伴也是給予我很多的幫助，大家互相勉勵，並督促彼此的進度，也在準備實驗時給予很多的意見和幫助，若沒有大家的幫忙，本篇論文就不會有機會完整呈現。

四年，一段很長的時間，我終於完成自己的夢想，完成了這一篇論文。雖然有些遲，雖然一路坎坷，但有機會受教於許多令人尊重的老師，有機會認識許多好夥伴，並取得學位，仍令人感動且開心。最後我要感謝我的妻子，一路以來，她相信我、鼓勵我、支持我，讓我能無後顧之憂的準備論文和實驗，直到我完成論文。

最後，僅在此感謝一路上支持我的老師、家人、夥伴和關心我的朋友們，因為有您們的支持與鼓勵，才讓我嚐到最終甜美的果實，謝謝您們！

目 錄

目 錄.....	I
圖 次.....	II
表 次.....	III
第一章 緒論.....	1
第一節 研究背景與動機.....	1
第二節 研究目的.....	5
第三節 名詞解釋.....	5
第二章 文獻探討	6
第一節 問題解決.....	6
第二節 程式設計學習.....	12
第三章 研究方法.....	16
第一節 研究設計.....	16
第二節 研究參與者.....	16
第三節 研究工具.....	17
第四節 教學規劃.....	23
第五節 實施步驟.....	26
第六節 資料分析方法.....	29
第四章 結果與討論.....	30
第一節 實驗組與控制組比較.....	30
第二節 性別的影響.....	31
第三節 分別比較男女實際成長.....	35
第五章 結論與建議.....	40
第一節 結論.....	40
第二節 建議.....	42
參考文獻.....	43
附錄A 機器車程式設計教學講義.....	49
附錄B 新編問題解決測驗.....	64

圖 次

圖2-1 Parnes和Noller於1967年提出的問題解決模式圖示（引自Firesteian & Treffinger,1983）	7
圖2-2 Isaksen、Puccio及Treffinger於1985年提出的六階段問題解決模式.....	8
圖2-3 Isaksen等學者於1989年提出的新問題解決模式.....	9
圖2-4 研究者歸納的問題解決模式.....	10
圖2-5 Deek等人與Isaksen等人的問題解決比較圖（資料來源：研究者整理）.	13
圖3-1 NXT主機與部件	17
圖3-2 NXT組裝成品	17
圖3-3 LEGO MINDSTORMS NXT software.....	18
圖3-4 機器車程式設計態度問卷.....	22
圖3-5 實驗實施步驟	26
圖3-6 實驗組學生上課情形.....	27

表 次

表 4-1 兩組全班問題解決能力 t 檢定結果.....	30
表 4-2 實驗組學生學習態度填答情形.....	32
表 4-3 實驗組對機器車程式設計的學習情形的填答狀況.....	33
表 4-4 實驗組學生學習態度填答情形 (男女)	34
表 4-5 實驗組對機器車程式設計的學習情形填答狀況 (男女)	35
表 4-6 實驗組男女問題解決能力 t 檢定結果.....	36
表 4-7 兩組男生問題解決能力 t 檢定結果.....	37
表 4-8 兩組女生問題解決能力 t 檢定結果.....	38

第一章 緒論

第一節 研究背景與動機

二十一世紀網路資訊化社會來臨，為了讓學生具備終生學習能力，前教育部部長 黃榮村（2003）的「教育部資訊教育辦理現況」專案報告裡提到，在資訊化的社會中，運用資訊科技主動學習與創新思考已成為各國教育發展的重點。我國九年一貫課程（1995）也指出，「國民教育階段學校教育目的，需引領學生致力達成十項國民教育基本能力」，其中便包含「運用科技與資訊的能力」、「培養獨立思考與問題解決的能力」。從上述不難發現，如「創新思考」及「問題解決能力」等高層次思考能力為現今教育的主要課題。

如何運用資訊教育課程以提升學生之高層次思考能力？Duke，Salzman，Burmeister，Poon及Murray（2000）認為學習程式設計能夠培養學生問題解決以及高層次思考能力，並有助於他們日後的職業規劃。程式設計在電腦科學領域之中扮演重要的角色，學習程式設計除了可習得運用程式解決問題之外，並進一步可以學習運用到其他領域的問題解決技巧（Linn, 1985）。Ellinger（2003）指出，程式設計也是一種表達的工具，讓小朋友可以透過程式語言的平台表達出自己的才華，並且，在學習程式設計的過程中，小朋友可以學習到電腦的運作方式；程式設計的學習能學到專業、展現個人風格以及和別人合作的能力。透過程式設計，學童們更是可以學到謹慎思維、自我批判、責任、創造力、溝通、合作、分享等多方面的知識以及概念。九十九年的高中「資訊科技概論」課綱中，資訊科技概論納入為高中必修課程，而程式設計的相關教學內容更佔電腦課程總時數的三分之一（教育部，2008）。由此可知，電腦是一項非常重要的課程，而程式設計更是電腦課程中的重點之一。Papert（1980）指出，學習程式設計能夠培養學習者的邏輯思考、問題解決及高階思考能力。雖有多位學者認同學習程式設計能提升學習者的問題解決能力，但程式設計與問題解決的關聯何在？曾有學者對程式設

計與問題解決間的關聯進行整理，提出程式設計問題解決歷程大致可以分為瞭解問題需求、擬定解題計劃、撰寫程式及測試與除錯等四個步驟（Deek, Kimmel & McHugh, 1998），明確顯示出程式設計與問題解決之間的關聯。

程式設計初學者在建構問題解決方法與瞭解程式執行過程時，通常要面對複雜的語法與指令，這些複雜的語法與指令，使得學習程式設計被認為是一項困難且複雜的工作（Kelleher&Pausch, 2005）。Beaubouef及Mason（2005）便指出，學生常在完成程式設計課程後僅能理解語法，卻無法自行解決問題以及進行程式規劃。Costelloe（2004）也指出學生初學程式設計時，在語法及程式設計概念上常會碰到困難並感到沮喪。程式設計初學者在傳統程式設計環境學習程式設計，當他們遇到錯誤時，往往會感到灰心喪志（Powers、Ecott & Hirshfield, 2004）。

Hadjrrouit（1998）表示，程式設計初學者學習C++的困難，在於程式設計的概念過於抽象，並且需要新的思維才能了解程式設計的內涵。程式設計的困難在於程式概念牽涉到過多的抽象化元素，因而造成教學者的負擔，也常使程式設計初學者感到挫敗（du Boualy, 1989）。程式設計初學者在學程式設計時遇到的困難，莫過於他們並沒有足夠的知識和技巧來建置一個完整的程式（Gandy, Bradley, Brookes & Allen, 2010）。電腦的思考模式和人的思考模式是不同的。程式設計初學者的認知模型建構於傳統的知識，認知的突然改變，造成初學者必須付出更多的努力才可以得到成長（Dale & Patricia, 2007）。

在這樣的前提之下，程式設計初學者往往會利用各種的方法去認識一個全新的語言，例如，許多學生會利用觀察輸出以及試誤的方式來檢驗程式的正確與否，但一般指令式程式語言的輸出卻只是一個簡單的畫面，造成程式設計初學者無法了解程式本身運作的過程。程式設計是一個循序漸進的過程，在電腦真正執行之前，心裡必須要先有一個虛擬的機器，去模擬運作的模型，才能預測執行的結果。但是因為學生心裡並沒有足夠的程式設計相關概念，因此往往會有許多迷思的產生。程式設計課程雖然占有重要的地位，但初學者在面對程式設計這樣的抽象概

念時，往往會出現許多困難，除了語意方面的問題，程式設計初學者還必須要面對語法上的錯誤，而這樣各種面相的問題交錯的結果，導致程式設計初學者在學習時，總是事倍功半，而老師在實際的教學過程中，也有種種難題需要克服。

許多學者發現，使用視覺化程式設計環境，有助於程式設計初學者在程式設計方面的學習。Du Boualy (1989) 指出，初學者遭遇到的最大問題在於程式設計的概念太過抽象、不易理解，他認為初學者的程式語言環境，應具有簡單化及視覺化等特性。視覺化程式設計環境的優勢在於它們可以以視覺化的方法表達程式運作的流程 (Dann, Cooper & Pausch, 2000)。Parsons 及 Haden (2007) 便提及，視覺化程式設計環境中，程式的運作流程可以使用視覺化的方式進行表達，學習者們不再需要在自己的內心去轉化程式的步驟、流程，如此學習者可以更有效的去建構出對於電腦行為的理解。視覺化程式設計環境提供學習者更具象化的介面，可幫助初學者掌握程式設計的架構 (Brusilovsky & Spring, 2004)，不像一般的程式語言語法結構龐大複雜，容易造成程式初學者學習的困擾 (Mannila, Peltomäki & Salakoski, 2006)。視覺化程式語言環境讓程式設計初學者較容易接受，因為那些會讓學生感到困惑的概念，像是程式實際運作的流程會被清楚的表達出來 (Parsons 及 Haden, 2007)。

由美國麻省理工學院 (MIT) 媒體實驗室與丹麥樂高 (LEGO) 公司合作開發的樂高機器車是利用樂高積木與電腦結合做為教具，進而發展出一系列的科學創意課程，研究發現，學生從實際動手做的過程中，能開拓學生的科學領域、邏輯分析及程式觀念的建立並激發學生對科學濃厚的興趣 (Sargent, Resnick, Martin & Silverman, 1996)。Stephen、Campbell 與 Ian (2007) 表示，樂高機器車是一個好的問題解決教材，因為它把程式運作的過程可視化。學生可以透過觀察機器車的整體運作流程來建構心中的認知架構。透過觀察，學生可以清楚的了解程式完整的運作流程，並且對程式進行修改，在這樣的流程下，學生可以一步步建置出內心對於程式設計的正确架構。Vassilios、Maya 及 Katerina (2005) 也表示，利

用機器車進行程式設計教學，能有效的提升學生的學習動機。國內許多學者將樂高機器車應用於中小學的程式設計教學並且獲得正面的成效（王正如, 2010；許雅慧, 2006；劉洲, 2005；謝亞錚, 2009）。王正如的研究發現，以機器為車前導組織有助於提升學生的程式設計能力，顯示機器車程式設計有助於學生建立程式設計之整體概念。機器車有助於學生在心中產生程式執行的具體心像，且學生較容易了解高階程式設計的指令（許雅慧, 2006）。謝亞錚（2009）表示，學生能接受機器車的程式指令、能透過機器車想像程式的執行結果，而且學生覺得機器車程式設計好玩、有成就感、挑戰性、可引起繼續學程式的興趣。由上述可知，使用樂高機器車進行程式設計教學有助於初學者對於程式設計的學習。

雖有多位學者及文獻支持以機器車進行程式設計教學，但比較機器車於程式設計教學時問題解決能力成長之研究卻實為有限，無法得知機器車運用於程式設計程式設計初學者問題解決能力的提升。因此，本研究希望探討學生使用機器車對於問題解決能力之提升，及學生對於此種學習方式的態度，以供教師及研究者在未來教學及研究的參考。

第二節 研究目的

本研究旨在探討以機器車做為程式設計教學工具對程式設計初學者問題解決能力之影響。本研究的主要目的有以下二點：

1. 探討以機器車進行程式設計教學對程式設計初學者問題解決能力之影響。
2. 探討以機器車進行程式設計教學對程式設計初學者學習程式設計態度之影響。
3. 探討性別對程式設計初學者學習機器車程式設計學習態度之影響。

第三節 名詞解釋

問題解決：本研究對問題解決之定義，為參考多位學者(Parnes, 1987; Isaksen, Puccio & Treffinger, 1993; Howe, 1997) 整理的問題解決模式後歸納而成。研究者將問題解決模式歸納為三個循環階段，包含「準備問題」、「激發點子」及「行動計畫」等三個循環階段。

第二章 文獻探討

第一節 問題解決

本節將透過文獻來探討什麼是問題解決，以及國內外相關問題解決評量工具。

壹、問題解決的定義

從十九世紀以來，問題解決被視為一種機械的、有系統的一套技巧，例如被用來解決數學問題，這些問題通常有一個合乎邏輯的正確答案，但在認知學習理論的影響下，問題解決轉換為包含多元認知技巧及行動的一種複雜心理活動。問題解決包含了高層次思考，如聯想、推論、理解、分析、綜合、歸納等技巧，且每個技巧都需被計畫及協調（Foshay & Kirkley, 1998）。而國內外多位學者也對問題解決的觀點提出不同的看法，例如Polya（1981）便認為問題解決的開始，是人們將注意力放在問題的細節上，當問題的癥結被注意到的時候；Mayer（1985）則認為問題解決為問題解決者發現問題與過去經驗之間的關係並採取行動的一連續過程。他提出問題解決有三項特徵：(1)問題解決是認知而不是從行為的推論；(2)問題解決是導致個體尋求解決的行為；(3)問題解決是加入了操作或運用過去經驗的過程。而國內學者（張春興，2001；黃春金，2010）也曾對問題解決加以定義，認為問題解決是一種運用自身知識、經驗解決問題的歷程。

Parnes（1987）則認為問題解決分為五個階段：發現事實（Fact-Finding）、發現問題（Problem-Finding）、發現點子（Idea-Finding）、發現解答

（Solution-Finding）、尋求可被接受的解答（Acceptance-Finding）。這種分階段解題模式，通常用來解決開放性問題，其特色在於，每一個階段都先有發散式思考，以免遺漏任何可能的答案，再有聚斂式思考，以增強解決歷程中思考的周密性，而解題者利用這五個階段來解決問題，以下為其五個階段的說明：

（1）在「發現事實」中，解題者可利用5W1H（Who、What、Where、When、

Why、How) 策略自問自答以找到收集資料的方向及與問題相關的資訊。

- (2) 在「發現問題」中，解題者要設法釐清問題，以得到一個有明確敘述的問題。
- (3) 在「發現點子」階段，解題者要能找出可能解決前一階段所得到問題的點子。
- (4) 在「發現解答」階段，解題者要評估前一階段的點子，並找出一個最恰當的點子做為問題解答。
- (5) 在「尋求可被接受的解答」階段，解題者必須計畫執行此點子，確定此點子是否有效，若無效則回到前幾個步驟，以得到另一個可能的解決方法。

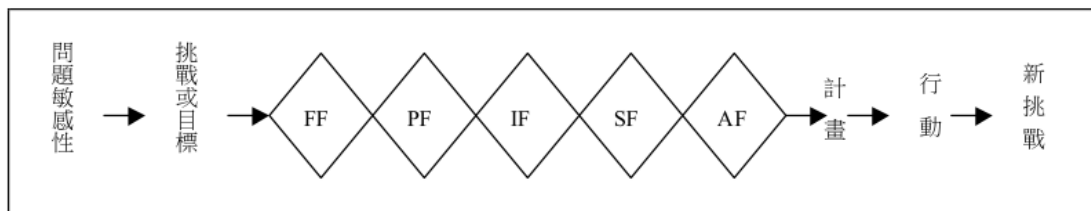


圖2-1 Parnes和Noller於1967年提出的問題解決模式圖示（引自Firesteian & Treffinger,1983）

1967年Parnes和其夥伴Noller以圖示法表達問題解決的過程（如圖2-1），其進展是建立在問題解決各階段中發散式與聚斂式的思考。然而Parnes所提出的問題解決模式，多注重發散式思考的訓練，而Isaksen、Puccio及Treffinger（1993）卻認為發散式與聚斂式的思考應並重，因此他們修改Parnes所提出的問題解決模式，加入聚斂式思考的觀點，提出新的六階段問題解決模式（如圖2-2）。其中主要修改重點如下：

- (1) 新增一個階段：「發現挑戰」（MF）。

(2) 將「發現事實」(FF)修改為「發現資料」(DF)。

(3) 發展聚斂式思考的指導綱領和技術，以平衡發散式思考。

而後，Isaksen等學者(Isaksen et al, 1993)發現人們在真實情境中解決問題時，會自然的將此六階段組合，並傾向於將此六階段分為「準備問題」、「激發點子」及「行動的計畫」三個部份，因此他在1989年提出新的問題解決模式，將原本的六個階段組合為三個部份，圖2-3為此新問題解決模式圖示。

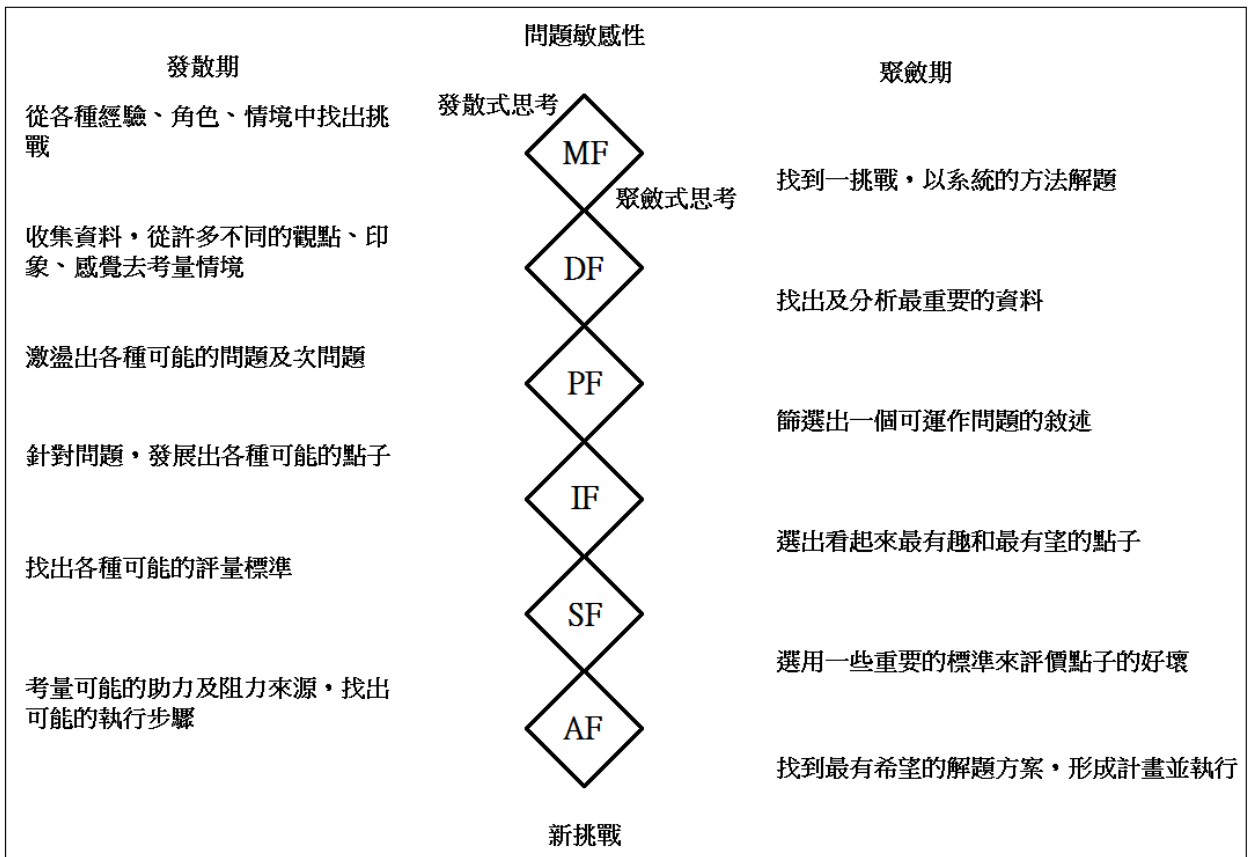


圖2-2 Isaksen、Puccio及Treffinger於1985年提出的六階段問題解決模式

MF	發現挑戰	準備問題 (getting the problem ready)
DF	發現資料	
PF	發現問題	
IF	發現點子	激發點子 (generating ideas)
SF	發現解答	行動的計畫 (planning for action)
AF	尋求接受	

圖2-3 Isaksen等學者於1989年提出的新問題解決模式

除此問題解決模式外，也曾有多位學者針對問題解決的定義及歷程提出不同的觀點，Howe（1997）便曾總結各種問題解決模式的特色，找出不同學者間的問題解決共通特色，包含：

- (1) 利用多階段方式循序達到解決問題的目的。
- (2) 每個階段都使用了發散式思考及聚斂式思考。
- (3) 每一階段都以發散式思考為開始，以聚斂式思考結束。
- (4) 問題解決適用於各人及團體。
- (5) 可使用一部分階段即可。
- (6) 各階段未必要按照一定的順序。
- (7) 各階段未必是一種線性呈現，也可成交互螺旋型呈現。

綜合以上觀點，研究者將問題解決模式歸納為三個循環階段，包含「準備問題」、「激發點子」及「行動計畫」等三個階段，每一階段均包含發散式思考及聚斂式思考，且皆可循環至較前階段，若在「行動計畫」結束後遭遇新問題，則循環至「準備問題」階段，其概念如圖2-4所示。

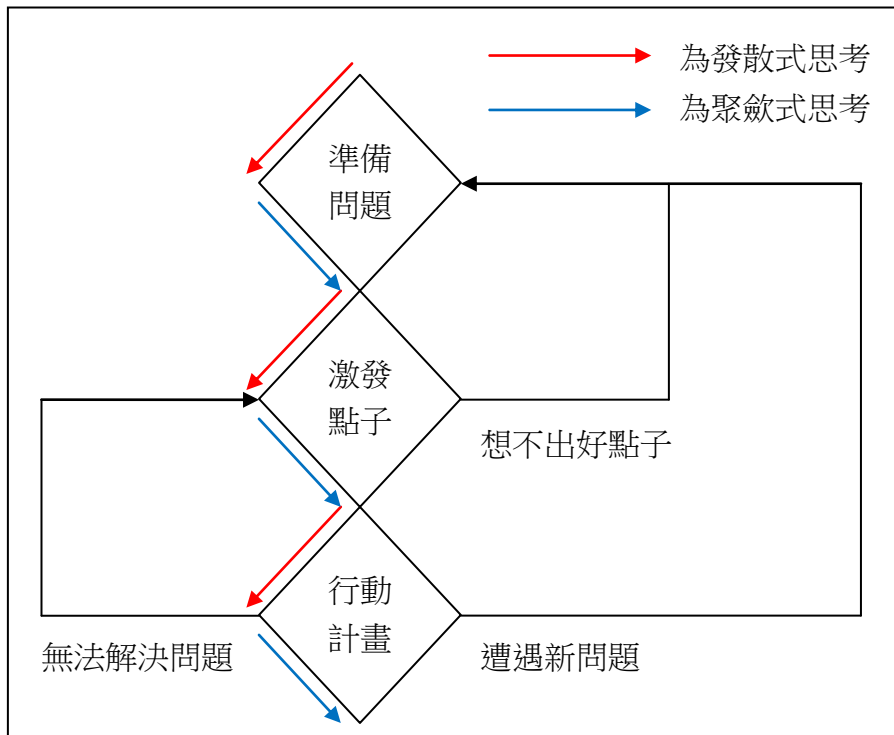


圖 2-4 研究者歸納的問題解決模式

貳、問題解決評量工具

根據美國教育部國家教育統計中心（National Center for Education Statistics, 2000）的報告指出，目前問題解決能力標準化評量被應用者包含：

- （1）COMP（ACT College Outcomes Measures Program problem solving subscale）：分為了解問題、了解脈絡知識、彙集可能的解決方案、定義及評估可能的限制、選擇方案及評估等六大步驟。
- （2）PSI（Problem Solving Inventory）：分為個體在問題解決時的自我認同、趨避傾向及自我控制等三項。
- （3）由Bowers、Huisingh、Barrett、Orman及LoGiudice編製而成的TOPS-E（Test of Problem Solving-Elementary）。

而國內的相關標準化測驗包含：

- （1）王萬清（1987）根據Guilford（1967）所提出的問題解決模式理論編

制而成的「問題解決能力測驗」，包含發現問題、界定問題、計畫實施步驟及提出可行方案等四個向度。

- (2) 詹秀美和吳武典(1991)修訂的「問題解決能力測驗」，是參考Zachman、Huising、Jorgensen及Barrett(1984)編製之「Test of problem solving」編製而成，包含解釋推論、猜測原因、逆向原因猜測、決定解決方法及預防問題等五項分測驗。
- (3) 羅芝芸(1999)的「問題解決能力測驗」是根據D`Zurilla和Nezu(1987)提出的問題解決理論及國內學者理論想法編製而成，包含學生逐步解決的能力、設法解決問題的能力、預想後果的能力、察覺問題存在的能力及察覺行為動機的能力等五個向度。
- (4) 李震歐(2001)編製的「科學問題解決測驗」，是以創造性問題解決為基礎，將測驗分為了解問題、激發構想及決定策略等三個階段，適用對象為國小三到六年級學生。
- (5) 詹秀美和吳武典(2002)參考原1991年的問題解決能力測驗，加入創造思考要素編製而成的「新編問題解決測驗」，其中包含界定原因、解決方法及預防問題等三個分測驗及變通性和有效性兩個向度。

以上五項標準化問題解決評量工具皆為參考國內外學者對問題解決的理論編製而成，並依據理論將問題解決分為三至五個向度，再針對此向度對評量加以編制。其中詹秀美和吳武典的「新編問題解決測驗」，修改後的評量內容包含變通性和有效性此二項度，其理論與發散式思考及聚斂式思考較為接近。

第二節 程式設計學習

本節將透過文獻來探討程式設計與問題解決的關聯，以及探討國內外相關運用樂高機器車於程式設計教學之案例。

壹、程式設計與問題解決

吳正己及何榮桂（1998）曾指出，為了達到解決問題的目標，程式設計課程在教學設計上應該「以問題解決為導向，再導入需要使用的相關指令與敘述」。而Winslow（1996）也同樣表示，程式設計的過程就是問題解決，程式語言教學的最終目標在於培養學習者透過設計程式來解決問題。Sieeman、Baxter與Kuapa（1986）表示，教程式設計的目的在於提升學生的解題能力以及邏輯性的思維。從上述可知，透過良好的教學策略及環境，學習程式設計可有效提升學生的問題解決能力，Feurzeig、Horwitz及Nickerson（1981）就曾以認知的角度，表示學習程式設計可以提升學童之嚴謹思考、除錯概念及提升問題解決能力。

曾有多位學者針對一般性問題解決提出他們的觀點（Isaksen et al, 1993；Hayes, 1980；Polya, 1981；Mayer,1992），但問題解決與程式設計的關聯何在？Newell及Simon（1972）曾由資訊處理之認知觀點提出「問題解決之資訊處理模式」，以問題的心智表徵與次目標構成之問題空間來說明問題解決的資訊處理過程。而Deek，Kimmel及McHugh（1998）更是將一般性問題解決歷程運用於程式設計領域，並提出程式設計問題解決歷程大致可以分為四個步驟：

- （1）瞭解問題需求：主要在釐清問題所包含的相關資訊，並瞭解與確認程式的輸入及輸出要求。
- （2）擬定解題計劃：將問題分解成幾個子問題，並以模組化的方式規劃出解決問題的步驟。
- （3）撰寫程式：撰寫程式以實行所設計的解題計劃。
- （4）測試與除錯：進行測試，確認解題的正確性並改正錯誤。

將Deek等人發表的程式設計問題解決歷程與Isaksen等人所發表的六階段問

題解決模式相比較，可以發現程式設計問題解決與一般性問題解決有極大的相似之處（如圖2-5）。然而，Schunk（2003）表示，在學習者熟悉的知識領域中，領域特定策略比一般性問題解決策略更為有效。表示在如程式設計這樣的特定領域中，學習者會用特定的方法進行學習，而非一般性問題解決模式。但Schunk也表示，若是在問題的解答不明顯的情況下，一般性問題解決策略則是很有用的尋求解答方法；而Andre（1986）也曾表示，一般性的問題解決模式，在解決不熟悉領域的問題時最為適用。

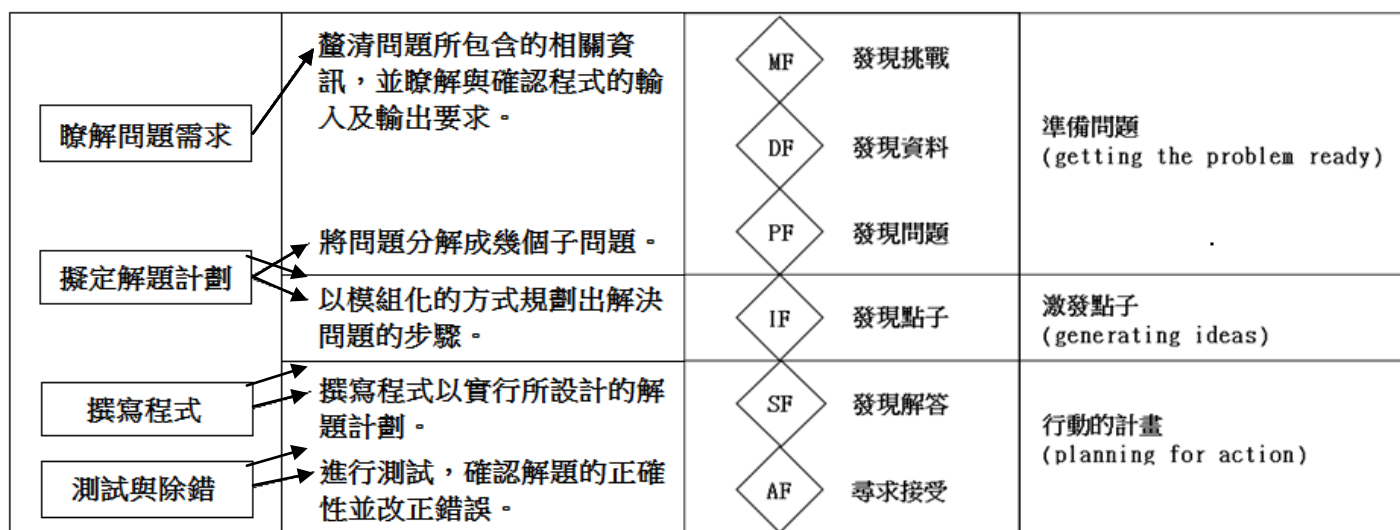


圖2-5 Deek 等人與Isaksen 等人的問題解決比較圖（資料來源：研究者整理）

綜合上述觀點，程式設計與問題解決的概念極為類似，兩者都是解決問題的歷程，並經過一定且相似的步驟。而雖然程式設計是一特定知識領域，學習者可能會依照過往的經驗而有不同的學習策略，但對於不熟悉程式設計的人，如程式設計初學者，則會應用一般性的問題解決模式來進行學習。

但電腦的思考模式和人的思考模式是不同的。程式設計初學者的認知模型建構於傳統的知識，認知的突然改變，造成初學者必須付出更多的努力才可以得到成長（Dale & Patricia, 2007）。程式設計初學者學習C++的困難，在於程式設計的概念過於抽象，並且需要新的思維才能了解程式設計的內涵（Hadjrrouit, 1998）。

Bruce及Foster (2004) 表示，學生在學習程式設計時，往往會對一些較抽象概念的理解產生困難。程式設計初學者在學程式設計時遇到的困難，莫過於他們並沒有足夠的知識和技巧來建置一個完整的程式 (Gandy, Bradley, Brookes & Allen, 2010)。在這樣的前提之下，程式設計初學者往往會利用各種的方法去認識一個全新的語言。許多學生會利用觀察輸出以及試誤的方式來檢驗程式的正確與否，但一般指令式程式語言的輸出卻只是一個簡單的畫面，造成程式設計初學者無法了解程式本身運作的過程。而在學習態度方面，對程式設計初學者來說，有趣、刺激可能比學習本身來的重要 (Malan & Leitner, 2007)。但程式設計初學者在傳統程式設計環境學習程式設計，當他們遇到錯誤時，往往會感到灰心喪志 (Powers, Ecott & Hirshfield, 2004)。

貳、機器車程式設計

田耐青(1999)認為樂高機器車程式設計是一個「科技支援之建構學習環境」，結合傳統的樂高積木與圖形化程式語言控制介面軟體，使得原本靜態的積木，產生更開放的使用空間。樂高機器車教學可提供學生問題解決的能力及合作學習的經驗，讓學生有正向的學習態度 (洪秋萍，2005)。

有多位學者使用樂高機器車進行研究，並發現使用樂高機器車於程式設計教學，能有效提升學生的學習興趣。Dagdilelis、Sartzemi及Kagani (2005) 認為樂高機器車因為真實、與科技連結、像玩具，因此利用樂高機器車進程式設計教學容易，能有效的提升學生的學習動機。Gandy、Bradley、Brookes與Allen(2010) 應用樂高機器車程式設計於初級程式設計的教學，並且在學生中取得廣大的迴響，學生甚至自主成立了機器車社。Norton、McRobbie與Ginns (2007) 利用Lego與其他科技搭配，並配合教師信念進行問題解決能力方面的研究，結果指出，樂高機器車這樣能豐富教學環境的教材，可以對學生的推理有正面的影響，並且能提升學習興趣。Meyer與Burhans (2007) 利用實體機器車以及模擬軟體進程式設

計的教學，先使用樂高機器車教導程式設計之基礎概念，之後再遷移至一般高階語言。結果顯示，學生認為使用機器車進行程式設計學習是有趣的，並且對於演算法的了解以及JAVA有正面的影響。

然而，雖有多位學者認為使用樂高機器車能有效提升學生學習程式設計的興趣，但Dambrot、Watkins-Malek、Silling、Marshall及Garver（1985）指出，男生的電腦學習成就比女性好。蔣姿儀（1997）的研究顯示電腦素養方面，男生的電腦素養，包含軟硬體、操作及應用等，均較女生好。楊中任及李榮通（2007）發現國小學生男生電腦學習成就優於女生；另一方面，Shashaani（1993）發現女性在電腦的學習是高於男性的，他以九至十二年級的高中生為對象，發現女生在班上的電腦成績較為優秀。透過以上研究，可以發現性別對學生學習電腦是有影響的，Kimberly（2002）以八年級為研究對象，發現使用不同教學方式，男女生電腦成就均有顯著差異，其中男生在使用電視教學後電腦成就優於女生，而女生在字卡教學後優於男生，Kimberly推測這其中的差異可能是興趣使然。Stein與Nickerson（2004）的研究也發現，在使用機器人進行教學時，女學生覺得有興趣的地方在於社交活動，如競賽、與他人比較，而男同學才是真正對機器人本身有興趣。綜上所述，性別因素可能會影響學生的學習態度，雖有多位學者認為使用樂高機器車能有效提升學生學習程式設計的興趣，但不同性別依然可能對機器車產生不同的興趣以及態度，因此性別因素是值得探討的地方。

第三章 研究方法

本研究主要是探討應用機器車輔助國中生學習程式設計對於學生學習之影響，檢驗在此種教學情境下是否能增進學生之問題解決能力及學習態度。

第一節 研究設計

本研究採準實驗研究法，研究分為實驗組及控制組二組。自變項為使用程式設計教學與否；依變項為學生之問題解決能力提升。

由於機器車數量有限，本研究將實驗組學生進行分組教學，每組二到三人。實驗時間為十週，每週授課二小時。第一週對實驗組及控制組參與者進行問題解決能力測驗前測，並對實驗組進行樂高機器車編譯環境之基本認識；第二到七週則為實際進行程式設計課程之時間，對實驗組學生進行樂高機器車之指令操作之教學。

第九週及第十週上半則進行競賽模式，根據研究者所設計之競賽地圖，參與者運用先前所學到之知識與其他組別同學進行小組競賽。第十週後半為問題解決能力測驗之後測以及態度問卷之填答。

控制組在實驗時間第一週進行問題解決能力測驗前測，並於實驗時間第十週進行問題解決能力測驗後測，其目的在於去除學生於實驗期間能力自然增長之可能性；實驗期間控制組進行資訊概論之教學。

第二節 研究參與者

本研究的研究參與者分為實驗組及控制組二個班級，其中實驗組男生14人、女生17人，共31人；控制組男生15人、女生17人，共32人。所有研究參與者均未曾學習過程式設計相關課程。

第三節 研究工具

本研究所使用之工具包括樂高機器車及其所附之拖曳式編譯軟體「LEGO MINDSTORMS NXT」軟體、程式設計教學講義、態度問卷、新編問題解決能力測驗。

壹、樂高機器車

本研究的實體機器車選用樂高機器車 (Lego Mindstorms NXT)。機器車上配備有觸碰感應器、光感應器、超音波感應器各一，以偵測外界環境狀況並反應。學生撰寫完程式後，就將機器車放置到實體環境中執行。圖3-1為NXT主機與部件連接之情形，透過感應器與馬達，可將NXT組裝為各式各樣的機器車（如圖3-2）。



圖 3-1 NXT主機與部件



圖 3-2 NXT組裝成品

貳、LEGO MINDSTORMS NXT 軟體

LEGO MINDSTORMS NXT 軟體為樂高機器車NXT內附之編譯軟體，使用者可透過拖曳的方式，拖曳其內建之功能區塊，例如前進、後退、轉彎、重複、感應器判斷等功能區塊，依照循序的方式對不同功能區塊進行排列、組合，以達到操縱樂高機器車之目的。圖3-3為LEGO MINDSTORMS NXT軟體實際運作畫面。

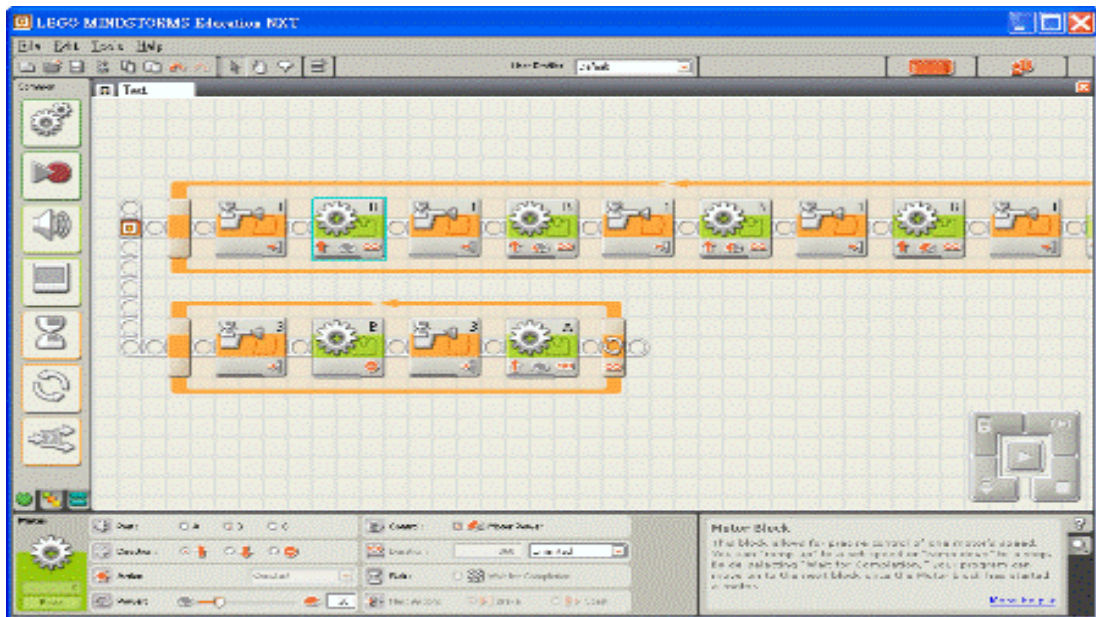


圖 3-3 LEGO MINDSTORMS NXT 軟體

參、程式設計教學講義

本研究使用的講義為機器車程式設計講義，講義僅作為輔助課程教學並歸類為參考用。講義上附有學生可能會用到的所有語法及技巧。

機器車程式設計教學講義用於實驗組，使用時間為第二週到第九週共十六小時的時間，講義上詳述所有課堂中會用到的操控機器車方法，以圖示加上敘述，並列舉各種程式撰寫技巧；在第四節教學規劃中研究者詳述各單元所教學的內容及概念，以及其與問題解決能力的相關處，實際講義內容請參見附錄A。

肆、機器車程式設計學習活動問卷

本問卷為參考國內相關文獻（王正如，2010；曾義智，2007；謝亞錚，2009）修訂完成，修訂後的問卷經由兩位專家修改確認後定稿。此問卷之目的在於學生在經過八週（十六小時）的機器車程式設計課程後，了解其學習情形以及學習態度。學習情形主要分為四大類，分為詢問機器車程式設計整體難易度（2題）、指令學習情況（1題）、是否有能力自行規劃程式（3題）、除錯能力（1題）；在學習態度方面，主要是想了解學生對於學習程式設計的想法以及是否具備繼續學習程式設計的意願（共9題）。

問卷發放的時間為實驗的最後一週（第十週）。問卷內容為學生學習程式設計的學習態度，採李克特式五點量表，填答項目有非常同意、同意、普通、不同意、非常不同意等五項，以藉此了解學生對機器車程式設計之學習態度。詳細問卷題目請參閱附錄B。

伍、問題解決能力測驗工具

本研究採用詹秀美和吳武典編制之「新編問題解決測驗」。問題解決測驗為詹秀美和吳武典（1991）參考Zachman、Huisinigh、Jorgensen及Barrett（1984）編製之「Test of problem solving」編製而成。並於2002年時，認為「創造思考」亦為問題解決能力的重要因素之一，於是將問題解決能力重新分類，其能力如下：

- (1) 界定原因：設想問題形成的原因之能力。
- (2) 解決方法：提供解決問題方法之能力。
- (3) 預防問題：預防問題再次發生之能力。
- (4) 變通性：提出具有變通性、富有彈性和獨創性之解決問題能力。
- (5) 有效性：提出有效解決問題之能力。

詹秀美和吳武典所編製的「新編問題解決測驗」和研究者所定義之問題解決模

式最為相關，其中界定原因對照準備問題、解決方法對照激發點子及行動計劃、預防問題對照新問題循環，而變通性及有效性則分別對照發散式思考及聚斂式思考；另其適用對象為四到七年級學生，符合研究者實驗對象為七年級學生之需求。

全測驗共包含六大題，每大題包含一段問題情境敘述和二至三個與問題情境相關的開放式問題，要求學生就每一個問題提出至多三項最好的答案；全測驗共有十五個問題，分屬於「界定原因」、「解決方法」與「預防問題」三個分測驗；此外，依據學生在每個問題之回答內容的多樣性與思考周密程度，評量學生解決問題思考之「變通性」與「有效性」。施測所需時間為50分鐘，三個分測驗的問題形式簡介如下所述。

分測驗一：界定原因

請學生針對問題情境之敘述內容，提出導致問題的各種可能原因，以評量學生從各種思考角度尋求問題可能原因的思考能力。

分測驗二：解決方法

請學生提出解決問題的各種可行方法，以評量學生提出多樣而有效之解決問題構想的思考能力。

分測驗三：預防問題

請學生提出有助於避免再度遭遇類似問題的各種可行方法，以評量學生吸取解決問題的經驗，並獲得各種有助於防範未然之道的後設認知思考能力。

兩個向度的評量原則簡介如下：

向度一：變通性

評量學生在每個問題提出的答案所包含之類別數量，作為學生解決問題思考之「變通性」的評量指標；學生在每個問題提出的答案包含的類別越多，則「變通性」的分越高。

向度二：有效性

評量學生在每個問題的答案內容之思考周密程度，作為學生解決問題思考之「有效性」的評量指標；學生提出的答案呈現越高度之思考周密性者，則「有效性」得分越高。

圖3-5為某位學生之填答情形，其中一（1）（表示第一大題第（1）小題）為「解決方法」分測驗，一（2）為「界定原因」分測驗，一（3）為「預防問題」分測驗。

一、晚上九點多，小華正在趕著寫作業時，突然間燈光熄滅，房間內一片漆黑。小華心裡想：「糟了！如果今天晚上睡覺前不能將作業寫完，明天上學會被老師處罰……」

(1) 小華現在應該怎麼辦才好？

答案1	到客廳去寫
答案2	明天早一點到學校寫
答案3	點蠟燭

(2) 燈光熄滅了，可能的原因有哪些？

答案1	停電
答案2	燈炮壞掉
答案3	電力不足

(3) 為了避免再發生同樣的情況，小華可以採取哪些方法？

答案1	換電炮
答案2	多買一些備用燈炮
答案3	請人來修

圖3-4 新編問題解決測驗填答範例

圖3-5中，該名學生一(1)之得分為「變通性」3分；「有效性」5分，因其中答案1為類別A：使用他處照明設備；而答案2為類別D：早起寫作業；答案三為類別C：使用備用、緊急照明設備。故其答案包含A、C、D三種類別，故此小

題其「變通性」得分為3分；在「有效性」方面，此三小題均完整描述其方法，並均為有效之方法，故每小題得分均為2分，其「有效性」得分為6分。故該名學生一（1）（解決方法）之得分為「變通性」分數 $\times 2$ + 「有效性」分數 $= 3 \times 2 + 6 = 12$ 分。

第四節 教學規劃

壹、實驗組

實驗組之教學分為兩個階段，第一階段為知識建立，第二階段為競賽模式。

階段一、知識建立

第一階段的目的是在於讓學生學習機器車程式設計的指令及概念，以建構其知識架構，並透過機器車程式設計解題流程，在過程中以引導的方式讓學生能了解題目、產生演算法，並以嚴謹的方式組合各種指令，最後達到解題的目的，以達到增進學生問題解決能力中「界定問題」、「解決問題」及「有效性」等三項能力。

單元一：樂高程式介紹

1-1樂高程式設計的步驟：介紹撰寫樂高機器車程式的二步驟，第一步為以拖曳式介面撰寫程式，第二步為傳送程式至機器車內並執行。

1-2NXT software的環境：以圖示方法說明拖曳式環境以及傳送程式至機器車的方法。

1-3機器車的外觀：以實際的機器車介紹機器車的外觀以及各感應器功能、位置。

單元二：機器車基本操作

2-1機器車會依照你的指令前進：讓學生以前進、轉彎、後退等指令撰寫程式，讓他們了解循序概念。

2-2機器車走迷宮（習題）：讓學生設計程式使機器車能走出一口字型迷宮，以了解其對機器車基本程式架構和循序概念的理解程度。

單元三：迴圈與判斷式

3-1重要觀念介紹：讓學生觀察機器車行走模式，並告知其可利用迴圈及判斷的方式，運用觸碰感應器來達到讓機器車能自動行走之目的。

3-2自動巡邏（例題）：綜合3-1所講解之觀念，帶領學生實作自動巡邏機器車。

3-3智能巡邏機器車（習題）：讓學生使用距離感應器，以設計不碰到牆壁便能自行轉彎的智能巡邏機器車。

單元四：光感應器與判斷式進階運用

4-1光感應器：講解光感應器的運用方式及其光感應數值所代表的意思。

4-2自動跟蹤機器車（習題）：讓學生設計一程式，使用光感應器讓機器車可以自動繞行黑色橢圓地圖前進。

單元五：樂高機器車綜合應用

5-1重要觀念介紹：教導學生不規則地圖的概念，並告知在漸層色地圖中，光感應器可以讀到不同的數值。

5-2多重條件控制的判斷式：讓學生了解判斷式內可以包含另一個判斷式，以達到多重判斷的目的。

5-3進階跟蹤（例題）：帶領學生設計一程式，使機器車能逆時針繞行漸層色不規則之地圖，以加強其對多重判斷概念的了解。

5-4進階跟蹤2（習題）：讓學生設計一程式，使機器車能順時針繞行漸層色不規則之地圖，以檢驗其對多重判斷概念的了解。

階段二、競賽模式

第二階段的目的是在於讓學生自行組合運用第一階段所學習到的知識及經驗，並透過比賽的方式，讓學生可以藉此了解到其他人的想法，加以借鑑，讓他們能

從不同的角度來思考問題，以豐富學生的「變通性」，並透過不斷的循環比賽，讓學生能吸取失敗的經驗，對機器車程式加以修改，以改進之前比賽所犯的錯誤，達到「預防問題」此項能力成長的結果。

機器車相撲大賽：在漸層黑線圍成的橢圓形地圖中，學生必須使用機器車與他組競賽，編輯程式以控制機器車將對方機器車推倒或推出黑線之外，其規則如下：

- 1.機器車先停止的組別輸
- 2.機器車完全超出黑線的組別輸
- 3.機器車先翻車的組別輸

貳、控制組

控制組課程為計算機概論教學，共進行18小時，教學內容分為「WORD基礎應用教學」、「PPT基礎應用教學」、「基礎網頁設計」及「益智小遊戲」等四部分。

WORD基礎應用教學：讓學生了解鍵盤按鍵的位置以及WORD中基礎功能應用，包含變更字體、插入圖片、版面配置、插入特殊符號及WORD繪圖等功能。

PPT基礎應用教學：讓學生了解PPT中基礎功能應用，包含簡易動畫製作、插入圖片、插入附註、插入超連結等功能。

基礎網頁設計：運用WOPOP，讓學生套用架構並設計一個人首頁，內容包含個人介紹、相簿及作品分享。

益智小遊戲：在課餘時間讓學生於電腦上玩益智小遊戲，例如暗棋、跳棋及機智問答等小遊戲。

第五節 實施步驟

本實驗時間共十週，每週授課時數為二小時，實驗實施步驟如圖3-5所示。

各步驟說明如下。

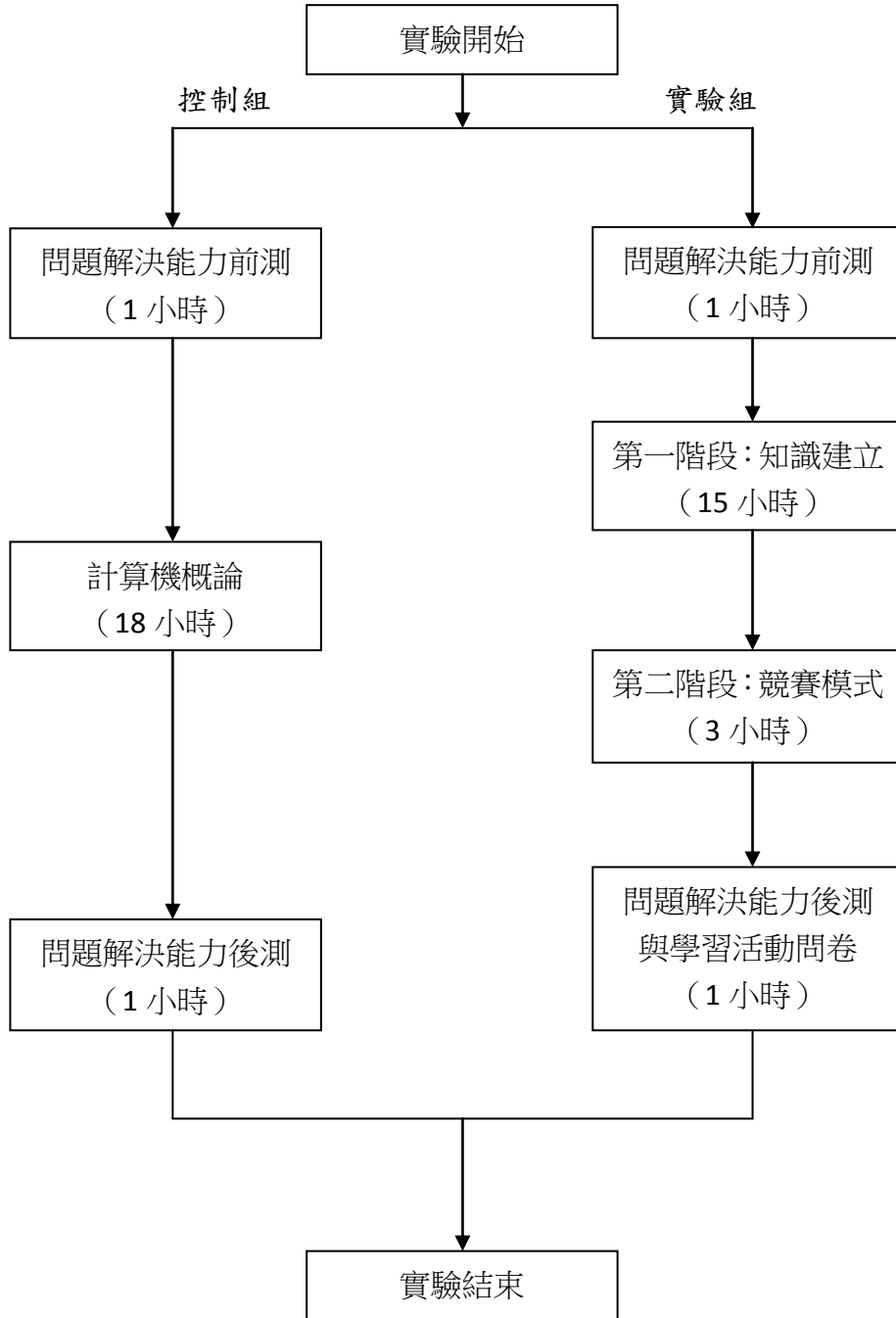


圖3-5 教學實驗實施步驟

壹、實驗組機器車程式設計教學

機器車程式設計教學共進行18小時，教學內容分為「第一階段-知識建立」及「第二階段-競賽模式」，其中「第一階段-知識建立」之目的在於將機器車程式設計之知識內容授予學生，而「第二階段-競賽模式」之目的則在於讓學生活用「第一階段-知識建立」中所學到的知識，並且透過競賽了解不同的思考、想法。圖3-6為實驗組學生進行機器車程式設計課程之情形。



圖3-6 實驗組學生上課情形

研究者在實驗組開始課程前，有先對另一班級進行教學，以對課程內容進行修正改進。由於樂高機器車程式設計環境是所有學生都沒有接觸過的，學生在剛接觸時難免會有些混亂並且有很多問題，因此在上課之前一定要把所有的軟體都成功安裝至電腦內，並且對樂高機器車進行韌體更新；第一次上課時不要先將機

器車分給各組同學，應先由老師對機器車進行完整介紹後再進行分配。若是有多位同學同時進行發問，老師可以先詢問每位學生的問題類型並給予提示，若多位學生的問題為類似概念，老師可以先請同學暫停手邊動作，並針對此概念為全班進行講解。

貳、控制組計算機概論教學

計算機概論教學共進行18小時，教學內容分為「WORD基礎應用教學」、「PPT基礎應用教學」、「基礎網頁設計」及「益智小遊戲」等四部分。其中「WORD基礎應用教學」主要為教導學生WORD文件的基礎操作使用，教學時間為4小時；「PPT基礎應用教學」為教導學生PPT投影片的基礎操作使用，教學時間為4小時；「基礎網頁設計」則是運用WOPOP，讓學生套用架構並設計一自己的首頁，教學時間為4小時；而「益智小遊戲」則是在每堂課的最後，會開放部分自由時間讓學生玩電腦的益智小遊戲，或是讓學生自由瀏覽網路、社交平台，時間約5小時。

參、學習活動問卷及問題解決能力測驗

實驗的第十週後半實驗組進行學習活動問卷及問題解決能力施測，控制組則進行問題解決能力施測。學習活動問卷之目的在於學生在經過八週（十六小時）的機器車程式設計課程後，了解其學習情形以及學習態度。學習情形主要分為四大類，分為詢問機器車程式設計整體難易度（2題）、指令學習情況（1題）、是否有能力自行規劃程式（3題）、除錯能力（1題）；在學習態度方面，主要是想了解學生對於學習程式設計的想法以及是否具備繼續學習程式設計的意願（共9題）；問題解決測驗使用詹秀美及吳武典編制之「新編問題解決測驗」，全測驗共包含六大題，每大題包含一段問題情境敘述和二至三個與問題情境相關的開放式問題，要求學生就每一個問題提出至多三項最好的答案；全測驗共有十五個問

題，分屬於「界定原因」、「解決方法」與「預防問題」三個分測驗；此外，依據學生在每個問題之回答內容的多樣性與思考周密程度，評量學生解決問題思考之「變通性」與「有效性」。

第六節 資料分析

資料分析主要以兩組之問題解決測驗及控制組之學習活動問卷作為分析重點，問題解決測驗以量化之方式進行統計分析，以了解兩組之間問題解決能力成長的差異，並經由學習活動問卷分析控制組男女之間問題解決能力成長之差異。

問題解決測驗分別對總分、界定原因、問題解決、預防問題、變通性及有效性等類別，將後測分數減去前測分數，採統計方法t檢定分析兩組學生在問題解決能力上的差異，並採 .05為顯著水準，當 p值小於 .05則表示兩組在該題達到顯著差異。

在態度問卷方面，本研究採李克特式五點量表，填答項目有非常同意、同意、普通、不同意、非常不同意，在分析時依序以5分、4分、3分、2分、1分來做為分析計分。

第四章 結果與討論

本章就機器車程式設計對初學者的問題解決能力提升及學習態度二方面來做結果討論。參與教學活動的人數為實驗組31人、控制組32人。

第一節 問題解決能力

表4-1為兩組實驗參與者在問題解決測驗進步分數之t檢定結果，在總成績分析上，結果顯示兩組成績並無顯著差異。 $(t=0.815, p=0.418)$ 。由於在「新編問題解決測驗」中，問題解決能力又可分為原因界定、解決方法、預防問題、變通性及有效性等五向度，因此，我們再進一步將問題解決能力分為此五個部分來做探討。然而在原因界定、解決方法、預防問題、變通性及有效性等五向度，表4-1的t檢定結果顯示兩組均無顯著差異，因此可發現，機器車程式設計對於整班學生問題解決能力表現雖有影響，卻並不顯著。

表4-1 兩組全班問題解決能力t檢定結果

問題解決能力	組別	人數	平均	標準差	t	p
總分	實驗組	28	13.64	12.708	.815	.418
	控制組	30	11.23	9.691		
界定問題	實驗組	28	3.93	5.340	.093	.926
	控制組	30	3.80	5.149		
解決方法	實驗組	28	4.71	5.662	1.832	.072
	控制組	30	2.57	2.932		
預防問題	實驗組	28	3.89	6.063	.705	.484
	控制組	30	4.93	5.166		
變通性	實驗組	28	2.54	2.701	.541	.591
	控制組	30	2.93	2.888		
有效性	實驗組	28	7.89	5.666	1.803	.077
	控制組	30	5.40	4.854		

* $p < .05$

由表4-1可以看出，實驗組與控制組兩組在「解決問題」與「有效性」二項雖未達顯著（解決問題 $p = .072$ ；有效性 $p = .077$ ），卻也十分接近，推測造成此二項未達顯著之原因有二：（1）實驗時間不足，實驗時間共十週二十小時，扣除前後測、態度問卷施測時間及機器車程式設計環境介紹，學生實際的上課時間約為十六小時，造成學生問題解決能力雖有成長，但因時間不足而未達顯著；（2）根據課堂觀察，在課程中，並非所有學生均對機器車程式設計課程持續抱有很大的學習熱誠，有部分學生的學習態度及學習情形較不理想，因而導致問題解決能力成長有限，此部分將於第二節中細加探討。

而在其它三向度「界定原因」、「預防問題」及「變通性」無顯著成長的原因，推測「界定原因」無顯著成長是因為教材中各章節太專注於該章節知識、指令、概念的教學，導致題型太過單一，例如單元三中例題與習題的問題均以「轉彎」為主，使學生在做習題時，可以直接參照例題之概念進行思考，造成學生沒有自行靈活思考機器車該怎麼走，從而造成「界定原因」此向度無顯著成長；「預防問題」無顯著成長原因，透過探討Deek, Kimmel及McHugh（1998）提出的程式設計問題解決歷程，其中包含瞭解問題需求、擬定解題計劃、撰寫程式及測試與除錯等四步驟，發現程式設計問題解決歷程中較缺乏預防問題發生的步驟，因此研究者推論學習程式設計較無法提升「預防問題」此項能力，進而造成無顯著成長；而「變通性」無顯著成長則是因為在教材中第二階段-競賽模式中，雖利用競賽的方式讓學生可互相參照他組機器車獲得勝利的方法，但教學中並未在競賽結束後實際讓各組呈現他們程式設計的內容，使學生無法實際了解他組設計概念，因此讓「變通性」此向度的成長有限。

第二節 學習態度

實驗組活動問卷分為學習情形及學習態度兩大類別，表4-2為實驗組學生對

樂高機器車程式設計課程之學習態度填答情，由表4-2我們可以看出，實驗組在學習態度方面每題平均得分都在五點量表的三分以上，顯示機器車程式設計課程對學生來說是好玩、輕鬆並具有挑戰性的。學生在學習機器車程式設計時，也大多保持認真的學習態度，認為能完成機器車程式設計很有成就感，也對往後學習程式設計課程保持較高的意願。由此可知，實驗組學生在面對機器車程式設計時，大多保持良好的學習心態，整體學習態度是正向的。

表4-2 實驗組學生學習態度填答情形

	非常 同意	同意	普通	不同意	非常不 同意	平均
1.我覺得機器車程式設計很好玩。	43%	14%	36%	7%	0%	3.93
3.我很認真的學習機器車程式設計。	14%	29%	25%	18%	14%	3.11
11.我覺得完成機器車程式設計很有成就感。	32%	29%	25%	14%	0%	3.79
12.我覺得機器車程式設計很有挑戰性。	43%	18%	32%	7%	0%	3.97
13.我希望學習更多的程式設計課程。	32%	11%	32%	21%	4%	3.46
14.我覺得學習機器車程式設計很輕鬆。	29%	21%	14%	29%	4%	3.33

表4-3為實驗組對機器車程式設計的學習情形的填答狀況，其中第二題及第九題表示學生對機器車程式設計的整體難易度體驗，第二題平均分數在五點量表未達三分，表示機器車程式設計對於學生有一定的困難度存在；但第九題平均分數在三分以上，卻又表示學生認為機器車程式設計並不會很抽象，配合第四題到第八題所呈現的結果，顯示學生能了解機器車程式的指令以及行動，當機器車行

動不如預期時，也能找出錯誤的地方，但唯獨第七題平均分數未達三分，顯示「組合機器車程式指令」對於學生來說是較困難的，因此研究者推測學生能了解機器車程式設計的概念、指令，也能透過思考想像出機器車該如何行動，但組合機器車程式指令對於學生卻較為困難。

表4-3 實驗組對機器車程式設計的學習情形的填答狀況

	非常 同意	同意	普通	不同意	非常不 同意	平均
2.我覺得機器車程式設計很容易理解。	0%	18%	50%	32%	0%	2.86
4.我能夠了解機器車程式的指令。	14%	36%	32%	18%	0%	3.46
5.我可以由閱讀機器車程式，想像出機器車可能的行動。	11%	29%	25%	29%	4%	3.08
6.給我題目，我就可以自己想像出機器車該如何行動才能達到目的地。	11%	21%	29%	36%	4%	3.02
7.給我題目，我不只可以自己想出解題步驟，還能自己組合出程式指令。	7%	14%	32%	39%	7%	2.72
8.若機器車行動不如預期，我可以自己找出錯誤的地方。	4%	36%	29%	29%	4%	3.13
9.我覺得機器車程式設計並不會很抽象。	7%	39%	36%	11%	7%	3.28

由於研究者在課堂觀察中發現，實驗組班級中男生的反應較女生熱烈，因此研究者對實驗組以性別為變因加以分析。

表4-4將實驗組中活動問卷中學習態度部分男生與女生的分數分開計算，顯示男生在學習態度的填答情形都較女生來的高，尤其女生在第三題、第十三題及第十四題平均得分均低於三分，顯示實驗組女生在機器車程式設計課程中學習較

不認真，學習機器車程式設計對女生來說並不是非常輕鬆，對於未來繼續學習程式設計相關課程的意願也較低。相較於男生，女生整體學習態度較低，這同時也呼應文獻探討中Stein及Nickerson（2004）所說「男生是真正對機器人感興趣，但女生的興趣卻在其他地方」。

表4-4 實驗組學生學習態度填答情形（男女）

	非常 同意	同意	普通	不同意	非常不 同意	平均
1.我覺得機器車程式設計很好玩。	男 64%	男 29%	男 7%	男 0%	男 0%	4.57
	女21%	女0%	女64%	女14%	女0%	3.25
3.我很認真的學習機器車程式設計。	男 21%	男 50%	男 7%	男 7%	男 14%	3.54
	女7%	女7%	女43%	女29%	女14%	2.64
11.我覺得完成機器車程式設計很有成就感。	男 43%	男 29%	男 29%	男 0%	男 0%	4.18
	女21%	女29%	女21%	女29%	女0%	3.42
12.我覺得機器車程式設計很有挑戰性。	男 50%	男 21%	男 29%	男 0%	男 0%	4.21
	女36%	女14%	女36%	女14%	女0%	3.72
13.我希望學習更多的程式設計課程。	男 50%	男 14%	男 29%	男 0%	男 7%	4
	女14%	女7%	女36%	女43%	女0%	2.92
14.我覺得學習機器車程式設計很輕鬆。	男 36%	男 43%	男 7%	男 7%	男 7%	3.94
	女21%	女0%	女21%	女57%	女0%	2.82

表4-5則將該班之學習情形填答分為男女分別計算，我們可以看出女生的平均分數普遍低於男生的平均分數，甚至在第二題、第五題、第六題、第七題、第八題等五題低於三分，顯示女生雖然能了解機器車程式設計的指令，但卻認為其不容易理解，並且較難想像出機器車是如何行動，也無法找出程式錯誤的地方。

配合上段學習態度，推測女生在學習時因較不認真，學習態度較差，因而導致學習情形降低，表示學習態度會影響學習情形。

表4-5 實驗組對機器車程式設計的學習情形的填答情形（男女）

	非常 同意	同意	普通	不同意	非常不 同意	平均
2.我覺得機器車程式設計很容易理解。	男 0%	男 29%	男 64%	男 7%	男 0%	3.22
	女 0%	女 7%	女 36%	女 57%	女 0%	2.50
4.我能夠了解機器車程式的指令。	男 21%	男 50%	男 21%	男 7%	男 0%	3.82
	女 7%	女 21%	女 43%	女 29%	女 0%	3.00
5.我可以由閱讀機器車程式，想像出機器車可能的行動。	男 14%	男 50%	男 29%	男 0%	男 7%	3.64
	女 7%	女 7%	女 21%	女 57%	女 0%	2.40
6.給我題目，我就可以自己想像出機器車該如何行動才能達到目的地。	男 14%	男 43%	男 29%	男 7%	男 7%	3.50
	女 7%	女 0%	女 29%	女 64%	女 0%	2.50
7.給我題目，我不只可以自己想出解題步驟，還能自己組合出程式指令。	男 7%	男 29%	男 43%	男 14%	男 7%	3.15
	女 7%	女 0%	女 21%	女 64%	女 7%	2.33
8.若機器車行動不如預期，我可以自己找出錯誤的地方。	男 7%	男 50%	男 21%	男 14%	男 7%	3.33
	女 0%	女 21%	女 36%	女 43%	女 0%	2.78
9.我覺得機器車程式設計並不會很抽象。	男 14%	男 36%	男 36%	男 7%	男 7%	3.43
	女 0%	女 43%	女 36%	女 14%	女 7%	3.15

第三節 綜合討論

由於對問卷分析的結果表示，實驗組中男生和女生的學習態度及學習情形是不同的，因此研究者以性別為變因，對實驗組問題解決能力進行統計檢定，將後測分數減去前測分數所得之差進行t檢定考驗，以分析其問題解決能力進步是否有差異，分析結果如表4-6。

表4-6 實驗組男女問題解決能力t檢定結果

問題解決能力	組別	人數	平均	標準差	t	p
總分	男生	11	15.91	9.669	.753	.458
	女生	17	12.18	14.432		
界定問題	男生	11	5.00	6.372	.850	.403
	女生	17	3.24	4.630		
解決方法	男生	11	5.18	4.423	.346	.732
	女生	17	4.41	6.452		
預防問題	男生	11	5.55	7.967	1.168	.253
	女生	17	2.82	4.391		
變通性	男生	11	3.09	2.879	.871	.392
	女生	17	2.18	2.604		
有效性	男生	11	10.64	5.316	2.203	.037*
	女生	17	6.12	5.290		

* $p < .05$

由表4-6可以發現，實驗組男同學在「有效性」此向度的成長顯著優於女同學，表示實驗組男同學在經過機器車程式設計教學後，思考變得更周密，並且更善於運用既有的知識與經驗，來掌握最具關鍵性的要素或最有效率的方法。配合活動問卷中學習態度與學習情形男女間的差異，推測良好的學習態度與學習情形，是為造成此差異的重要因素。程式設計學習態度影響學習情形，而程式設計學習情形則影響問題解決能力的成長。

表4-6顯示出不同性別在經過機器車程式設計課程後，其問題解決能力成長的差異，但卻不能從中了解各別的實際成長，因此研究者再將男女分開，將其結果與控制組加以比較，以了解其實際成長。

表4-7為實驗組及控制組兩組男同學在問題解決測驗的t檢定結果，將後測分數減去前測分數所得之差進行t檢定考驗，以測驗其問題解決能力進步是否有差異。在總成績分析上，結果顯示兩組總成績並無顯著差異。 $(t=0.902, p=0.376)$ 。

將問題解決能力分為原因界定、解決方法、預防問題、變通性及有效性等五向度分別分析探討，發現在解決方法及有效性等二向度顯示達顯著差異（解決方法 $p < .05$ ；有效性 $p < .05$ ）。表示機器車程式設計能有效提升男學生問題解決能力中「解決方法」及「有效性」等二向度表現。

表4-7 兩組男生問題解決能力t檢定結果

問題解決能力	組別	人數	平均	標準差	t	p
總分	實驗組	11	15.91	9.669	.902	.376
	控制組	15	11.73	12.892		
界定問題	實驗組	11	5.00	6.372	.057	.955
	控制組	15	4.87	5.436		
解決方法	實驗組	11	5.18	4.423	2.514	.019*
	控制組	15	1.60	2.849		
預防問題	實驗組	11	5.55	7.967	.093	.927
	控制組	15	5.27	7.294		
變通性	實驗組	11	3.09	2.879	.112	.912
	控制組	15	2.93	3.955		
有效性	實驗組	11	10.64	5.316	2.095	.047*
	控制組	15	5.93	5.885		

* $p < .05$

表4-8為實驗組及控制組兩組女同學在問題解決測驗的t檢定結果，將後測分數減去前測分數所得之差進行t檢定考驗，以測驗其問題解決能力進步是否有差異。在總成績分析上，結果顯示兩組總成績並無顯著差異。 $(t=0.366, p=0.717)$ 。將問題解決能力分為原因界定、解決方法、預防問題、變通性及有效性等五向度分別分析探討，顯示其結果雖有影響，卻均無達顯著差異。表示機器車程式設計雖對女同學問題解決能力成長有一定的影響，卻並不顯著。

表4-8 兩組女生問題解決能力t檢定結果

問題解決能力	組別	人數	平均	標準差	t	p
總分	實驗組	17	12.18	14.432	.366	.717
	控制組	15	10.73	5.271		
界定問題	實驗組	17	3.24	4.630	.301	.765
	控制組	15	2.73	4.788		
解決方法	實驗組	17	4.41	6.452	.488	.629
	控制組	15	3.53	2.774		
預防問題	實驗組	17	4.60	1.352	1.503	.143
	控制組	15	2.82	4.391		
變通性	實驗組	17	2.18	2.604	-1.021	.315
	控制組	15	2.93	1.280		
有效性	實驗組	17	6.12	5.290	.766	.450
	控制組	15	4.87	3.681		

*p<.05

由表4-7及表4-8我們可以發現，實驗組男同學在問題解決能力中「解決方法」及「有效性」等二向度有顯著提升，實驗組女同學在各向度則無顯著提升，配合學習活動問卷所發現，實驗組男同學在學習態度與學習情形均優於實驗組女同學，推測學生在良好的程式設計學習態度與學習情形下，能有效提升問題解決能力中「解決方法」及「有效性」等二向度能力成長。「解決方法」此向度之所以有顯著成長，是因為整個教學過程均著重於「該如何讓機器車依照指示行動」，因此學生在課程中不斷在思考該如何搭配、組合指令，使機器車依照自己的想法前進，而就是這樣的思考過程，有效增進學生「解決方法」此向度成長；「有效性」此向度顯著成長的原因則在於，教材中「第一階段-知識建立」，每單元教授指令及概念後，便要求其運用知識來進行例題與習題，而「第二階段-競賽模式」的設計理念，就是運用第一階段所學到的所有知識，符合「有效性」的解釋：「善於運用既有的知識與經驗，來掌握最具關鍵性的要素或最有效率的方法」，從而

造成「有效性」此向度顯著成長。

第五章 結論與建議

學習程式設計的過程與問題解決的歷程有許多相似之處，就如同Hunt(1994)所說的：「問題解決是個體在問題產生後，會進行問題解決的認知，再利用既有的知識與技能去服和新情境的需要以獲得解答的過程。」問題解決就是利用既有的知識與技能來解決問題；而程式設計的課程正是先將指令、語法等工具教給學生後，再讓他們利用所學來解決問題的一種歷程。本研究旨在探討機器車程式設計教學課程是否有助於提升學習者的問題解決能力，以下針對本研究結果歸納出結論，並根據實驗進行過程所發現到的問題提出建議，以做為未來相關研究之借鏡。

第一節 結論

壹、 機器車輔助程式設計對學生之問題解決能力無顯著影響

雖然在整體問題解決能力及其五項分項來看，實驗組與控制組之間並無顯著差異，但經由統計分析結果可以發現，其中「解決問題」與「有效性」此二向度結果已接近顯著（解決問題 $p = .072$ ；有效性 $p = .077$ ），表示機器車程式設計課程對學生此二向度能力成長仍具有一定的影響力，研究者推測因實驗時間不足及部分學生對機器車程式設計課程的學習態度及學習情形較不理想而造成此結果。另在其它三向度中「界定原因」無顯著成長的原因，推測因教材中題型太過單一而造成；「預防問題」無顯著成長的原因，透過比較程式設計問題解決與共通性問題解決之間的差異，發現在程式設計問題解決中較缺乏與此能力相關的步驟及過程，因此造成此能力無顯著成長；「變通性」無顯著成長則是因為在教材中第二階段-競賽模式中，雖利用競賽的方式讓學生可互相參照他組機器車獲得勝利的方法，但教學中並未在競賽結束後實際讓各組呈現他們程式設計的內容，使學生無法實際了解他組設計概念，因此讓「變通性」此向度的成長有限。

貳、 女性學生對機器車程式設計較無興趣

由學習活動問卷可以明顯看出，實驗組學生雖整體對於機器車程式設計均持正向態度，但若將男女分數分開，分別檢視其分數，可以發現實驗組男同學的學習態度與學習情形分數較高於實驗組女同學，顯示女生在學習機器車程式設計時較不認真，並且機器車程式設計對於女生有一定的難度。Stein及Nickerson(2004)說「男生是真正對機器人感興趣，但女生的興趣卻在其他地方」。因此研究者推測女生僅是對機器車此項工具較無興趣，而非對程式設計無興趣。

參、 男性學生對機器車程式設計有較高的興趣，在實驗後，其「解決方法」及「有效性」二個問題解決面向之能力有顯著提升

由於實驗組中，透過問卷及課堂觀察發現男同學的學習態度與學習情形較優於女同學，因此將男女分數分開，單獨看實驗組與控制組男生之間的比較，發現在「解決方法」及「有效性」此二向度，實驗組男同學顯著高於控制組男同學。其中「解決方法」分數較高，表示實驗組男同學在經過機器車程式設計教學後，能提出更多樣而有效之解決問題構想；而「有效性」分數較高，表示實驗組男同學在經過機器車程式設計教學後，思考變得更周密，並且更善於運用既有的知識與經驗，來掌握最具關鍵性的要素或最有效率的方法。由於研究者在進行教學時，主要以引導的方式來讓學生自行思考機器車該如何行動才能達到目的地，或該如何組合指令才能讓機器車行動如預期，因此研究者認為，這樣的課程可以讓學生更謹慎的去思考、去了解、去表達出自己想表達的想法，從而讓其思考更周密，也更能掌握並運用自己所擁有的知識及經驗。

第二節 建議

以下根據本研究的實施過程與結果，對未來相關研究提出建議。

壹、建議重覆本研究，但增加其它程式設計環境進行比較

不同的工具應該會對能力的成長有不同的影響，因此研究者建議在未來的研究中，可以加入不同的程式設計環境如Scratch、Alice等環境互相比較，以探討不同程式設計環境對初學者問題解決能力的影響。

貳、增加輔導員

由於研究參與人員除研究者本人擔任教學老師外，僅有一名輔導員，而大多數的學生在課程的過程中都會產生問題，更因為研究者為讓課程內容更符合問題解決歷程，當學生發生問題時，研究者多以引導方式來為學生解答；但僅有2名問題解決者較難顧全整班約12組30名學生，因此研究者建議增加輔導員至3名，加上教學者共4名問題解決者於約12組30人之班級，平均一人負責3組學生較為適當。

參、加強質性研究

由於本研究中研究者僅以問卷之方法蒐集質性資料，因此對於質性部分較無深入探討，例如在態度問卷中，女同學的學習興趣較男生低，但研究者卻無法深入了解女同學為什麼學習興趣會比較低，也無法知道為什麼男同學會對機器車程式設計抱有如此大的熱情。研究者認為這是一個很值得探討的議題，去真正了解性別變因對機器車程式設計的影響，因此建議加強質性研究，以探討其真正的原因所在。

參考文獻

- 王正如 (2010)。機器車前導組織對程式設計心智模型之影響。國立台灣師範大學輔導研究碩士論文。
- 王萬清 (1987)。電腦輔助問題解決課程對兒童問題解決能力及程序思考能力之影響。國立台灣師範大學輔導研究碩士論文。
- 田耐青 (1999)。由「電腦樂高」談新世紀的學習：一個「科技支援之建構學習環境」實例。教學科技與媒體，第44期，24-35頁。
- 李震歐 (2001)。高低學習成就學童科學問題解決能力之比較研究。國立屏東師範學院數理教育研究所碩士論文。
- 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。
- 黃春金 (2010)。探索教育活動對國小高年級學童的問題解決能力與態度之影響研究。屏東教育大學碩士論文。
- 教育部 (1995)。高級中學課程標準。台北：教育部。
- 教育部 (2003)。教育部資訊教育辦理現況。台北：教育部。
- 許雅惠 (2006)。應用LEGO Mindstorms 視覺化環境輔助程式設計觀念學習。國立臺灣師範大學資訊教育學系碩士論文。
- 張春興 (2001)。現代心理學。台北：東華書局。
- 曾義智 (2007)。應用機器人於程式設計教學—實體機器人與模擬軟體使用成效比較。國立台灣師範大學資訊教育研究所碩士論文。
- 楊中任、李榮通 (2007) 九年一貫資訊教育成效之探討—以中部地區國小高年級學童為例。國立嘉義大學國民教育研究所碩士論文。
- 詹秀美、吳武典 (2002)。新編問題解決測驗。台北市：心理出版社。
- 劉洲 (2005)。應用LEGO Mindstorms 在高中程式設計教學的成效探討。國立臺灣師範大學資訊教育研究所碩士論文。

- 蔣姿儀 (1997)。國民中小學學生電腦態度、電腦素養及其相關因素之研究。國立政治大學教育研究所。
- 謝亞錚 (2009)。機器人輔助程式設計教學之學習成效與學生心智模型探討。國立臺灣師範大學資訊教育學系碩士論文。
- 羅芝芸 (1999)。兒童認知風格、情緒智力與問題解決能力之相關研究。國立高雄師範大學教育研究所碩士論文。
- Andre, T. (1986). Problem and education. In G. D. Phye & T. Andre (Eds.), *Cognitive classroom learning: Understanding, thinking, and problem solving*, 169-204. Orlando: Academic Press.
- Beaubouef, T. & Mason, J. (2005) Why the High Attrition Rate for Computer Science Students: Some Thoughts and Observations. *ACM SIGCSE Bulletin*, 37 (2), 103-106.
- Bruce, K. B. & Foster, J. N. (2004). LOOJ: Weaving LOOM into Java. *ECOOP 2004: Object-Oriented Programming*. SpringerVerlag, Oslo, Norway.
- Brusilovsky, P., & Spring, M. (2004). Adaptive, engaging, and explanatory visualization in a C programming course. In L. Cantoni & C. McLoughlin (Eds.), *Proceedings of World Conference on Educational Media, Hypermedia, and Telecommunications 2004*, 1264-1271. Chesapeake: VA: AACE.
- Carlson, P.& Bremmer, D. (2007).Transforming middle-school curricula to reflect the new IT literacies of 21st century STEM careers. *American Society for Engineering Education National Conference, Honolulu, HI*, 22 – 25.
- Cooper, S., Dann, W., and Paush, R. (2000). Alice: a 3-D tool for introductory programming concepts. *Journal of Computing in Small Colleges*, 15(5), 108-117.
- COSTELLOE, E. (2004) Teaching Programming The State of the Art. *Department of Computing, Institute of Technology Tallaght, Dublin 24. CRITE Technical*

Report.

- Dagdilelis, V., Sartatzemi, M. & Kagani, K. (2005). Teaching (with) robots in secondary schools: some new and not-so-new pedagogical problems. *Proceedings of Fifth IEEE International Conference on Advanced Learning Technologies*. 757-761.
- Dambrot, F. H., Watkins-Malek, M. A., Silling, S. M., Marshall, R. S., & Garver, J. A. (1985). Correlates of sex differences in attitudes toward and involvement with computers. *Journal of Vocational Behavior*, 27, 71-86.
- Deek, F. P., Kimmel, H. & McHugh, J. A. (1998). Pedagogical changes in the delivery of the first-course in computer science: Problem solving, then programming. *Journal of Engineering Education*, 87, 313-320.
- Du Boulay, B., O'Shea, T., & Monk, J. (1989). The black box inside the glass box: presenting computing concepts to novices. In E. Soloway & J.C. Spohrer (Eds.), *Studying the novice programmer*, pp. 431-446.
- Duke, R., Salzman, E., Burmeister, J., Poon, J., & Murray, L. (2000). Teaching programming to beginners –choosing the language is just the first step. *Proceedings of the Australasian conference on Computing education*, 79-86.
- Ellinger, H. (2003). A natural sense of algorithm: Children should learn computer programming as a basic skill. Retrieved December 18, 2007, from <http://hunter.ellinger.org/education/ProgrammingForChildren.htm>
- Feurzig, W., Horwitz, P. & Nickerson, R. S. (1981). *Microcomputers in education*. Report No. 4798.
- Firestien, R. L. & Treffinger, D. J. (1983). Creative Problem Solving: Guidelines and Resources for Effective Facilitation. *Gifted Child Today*. January/February, 2-10.

- Foshay, R. & Kirkley, J. (1998). Principles for Teaching Problem Solving.
<http://www.plato.com/whitepapers.asp>
- Gandy, L., Bradley, S., Brookes D. A.& Allen, N. (2010)The use of LEGO Minstroms NXT Robots in the Teaching of Introductory Java Programming to Undergraduate Students.*ITALICS Volume 9 Issue 1 February 2010 ISSN: 1473-7507*, 1-9.
- Grandell, L., Peltomäki, M., Back, R. J. & Salakoski,T. (2006). Why complicate things?: introducing programming in high school using Python.*ACE '06 Proceedings of the 8th Australasian Conference on Computing Education*, 52, 71-80.
- Hayes, J. R. (1980). *The complete problem solver*. Philadelphia: Franklin Institute Press.
- Isaksen, S. G., Puccio, G. J. & Treffinger, D. J. (1993). An Ecological Approach to Creativity Research: Profiling for Creative Problem Solving. *The Journal of Creative Behavior*, 27(3), 149-169.
- Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming. *ACM Computing Surveys (CSUR)*, 37(2), 83-137.
- Kimberly, V. H. (2002). Gender Differences in Computer Technolgy Achievement. *Middle School Computer Technologies Journal a service of NC State University*, Raleigh, 5(2).
- Malan, D., & Leitner, H. (2007). Scratch for Budding Computer Scientists. SIGCSE Proceedings.
- Mayer, R. E. (1992). Teaching for transfer of problem-solving skills to computer programming.In E. D. Corte, M. C. Linn, H. Mandl & L. Verschaffel(Eds), *Learning environment and problem solving*, 193-206. NY: Spring-Verlag.

- Mannila, L., Peltomaki, M., & Salakoski, T. (2006). What about a simple language? Analyzing the difficulties in learning to program. *Computer science education*, 16(3), 211-227.
- Meyer, R. M., & Burhans, D. T. (2007). Robotran: A Programming Environment for Novices Using LEGO Mindstorms Robots. *Proceedings of the 20th International Florida Artificial Intelligence Research Society (FLAIRS) Conference*, 321-326.
- Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.
- Norton, S. J., McRobbie, C. J., Ginns, I. S. (2006). Problem Solving in a Middle School Robotics Design Classroom. *Research in Science Education* 37(3), 261-277.
- Papert, S. (1980c). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Parnes, S. J. (1987). Visioneering - State of the Art. *The Journal of Creative Behavior*, 21(3), 283-299.
- Parsons, D., & Haden, P. (2007). Programming Osmosis: Knowledge Transfer from Imperative to Visual Programming Environments. In S. Mann & N. Bridgeman (Eds.), *Proceedings of The Twentieth Annual NACCQ Conference*, 209-215.
- Polya, G. (1981). *Mathematical Discovery: On Understanding, Learning and Teaching Problem Solving*. New York: Wiley.
- Powers, K., Ecott, S., & Hirshfield, L. (2007). Through the looking glass: teaching CS0 with Alice. *Proceedings of the 38th SIGCSE technical symposium on Computer science education SIGCSE '07*, 39 1, 213-217.
- Putnam, R. T., Sleeman D., Baxter, J. A., & Kuspa, L. K. (1986). A summary of

- misconceptions of high school basic programmers. *Journal of Educational Computing Research*, 2(4), 459-472.
- Resnick, M., Martin, F., Sargent, R. & Silverman, B. (1996). Programmable Bricks: Toys to Think With. *IBM Systems Journal*, 35(3-4), 443-452.
- Satratzemi M., Dagdilelis V., & Kagani K. (2005). Teaching Programming with robots: A case Study on Greek Secondary Education. P. Bozani, E.N. Houstis, (Ed.), pp. *Lecture Notes in Computer Science (LNCS)*, Vol. 3746, 502-512.
- Schunk, D. H. (2003). *Learning theories: An educational perspective (4th ed.)*. New Jersey: Prentice-Hall.
- Shashaani, L. (1993). Gender-based differences in attitudes toward computers. *Computer and Education*, 20(2), 169-181.
- Smith, D. C., Cypher, A., & Tesler, L. (2000). Novice programming comes of age. *Communications of the ACM*, 43(3), 75-81.
- Stein, C., & Nickerson, K. (2004). Botball Robotics and Gender Differences in Middle School Teams. *Proceedings of the 2004 American Society for Engineering Education Annual Conference*.
- Winslow, L. E. (1996). Programming pedagogy: A psychological overview. *SIGCSE Bulletin*, 28, 17-22

單元一 樂高程式介紹

1-1 樂高程式設計的步驟

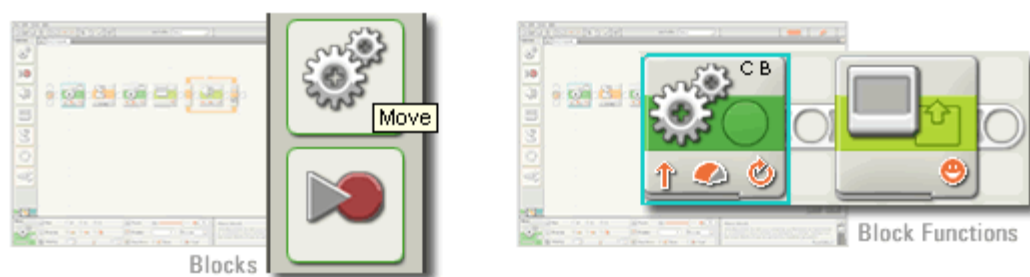
我們首先要知道撰寫樂高程式的二個步驟：

1. 撰寫程式：在此我們以 NXT Software 來撰寫樂高程式，利用拖曳的方法將物件拉入視窗並加以執行。
2. 將程式傳輸到機器車執行：在程式正確撰寫完程後就可以將程式傳到機器車上執行。

1-2 NXT Software 的環境

拖曳式環境

可將畫面左側的區塊拖曳至程式圖中。各個區塊將執行專屬的函式，如移動馬達、顯示訊息、偵測聲音，或量測距離。在整合一系列的區塊之後，即可隨心所欲讓機器人進行相關動作！



下載並執行

只要撰寫程式完畢，即可下載至 NXT。透過無線藍芽功能或 USB 連接線，即可將編譯完成的程式傳送至 NXT。NXT 將接著執行從主機電腦所接收的程式碼。

使用者可建立多組程式，並下載至自己的 NXT 機器人。



1-3 機器車的外觀

學習樂高程式設計的第一步，就是要知道機器車的構造。我們使用的機器車，有一個主機我們寫的程式可以儲存在這裡面，另外還有用來帶動機器車前進後退的馬達、觸碰感應器、距離感應器、聲音感應器以及光感應器等等。

我們可以看到機器車如下圖所示：



單元二 機器車基本操作

2-1 機器車會依照你給的指令前進！

程式的內容次序與程式執行的結果有密切的關係，不同的次序會導致不同的執行結果，會依照其程式內容依序執行之，而機器車程式執行的方式即是依序由左至右執行。



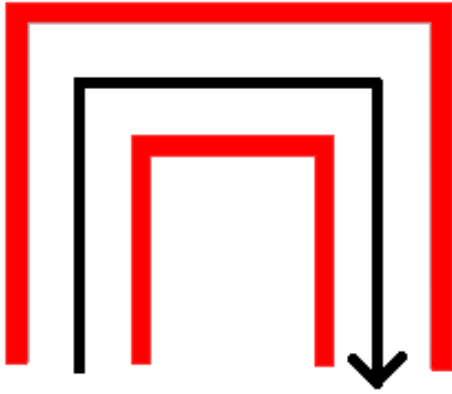
例如：

這時機器車就會依序先前進 5 秒，然後左轉 90 度，再前進 5 秒，最後再右轉 90 度。

2-2 機器車走迷宮

習題：

請設計一個程式使機器車能以順時針方向，走出如下圖所示的門字型迷宮。
(注意：請自行預估機器車要行走的距離。)



請問機器車要如何行進才能達到目的地呢？

根據上一題，你認為要用哪些指令區塊組合才能讓機器車達到目的地？請在下方畫出：

單元三迴圈與判斷式

3-1 重要觀念介紹

回想上一個單元，我們在習題二試著要走出門字形的迷宮時，還必需要量前進的距離，或者是要猜測轉彎的角度，如果迷宮的長度一變動，就必須要重新量測距離，而且一但碰到牆壁，機器車就可能卡死在迷宮內。那時的機器車還不夠聰明，沒有辦法知道是不是撞到東西了。為增加機器車的功能，讓機器車聰明一些本單元要介紹「觸碰感應器」以及搭配指令。

在介紹程式碼之前，我們先觀察一下有了觸碰感應器之後，機器車走迷宮的方式：

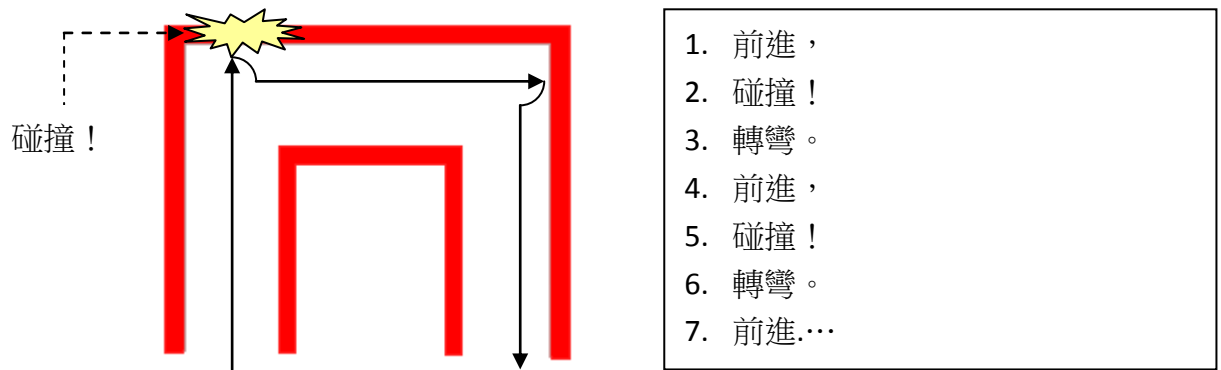


圖 3-1 觀察機器車的行為

在此，機器車會一直重覆前進、轉彎的兩種動作，而會進行轉彎的動作，是因為從觸碰感應器偵測到了一個「碰撞！」訊息，表示機器車已經碰到障礙物，必需進行轉彎的動作。

在這邊有兩個重要的概念：

第一個是「不斷重覆」的概念，這在程式當中就是迴圈。在本單元中，我們會用到的是迴圈。而在 NXT Software 中的迴圈應用方式為：

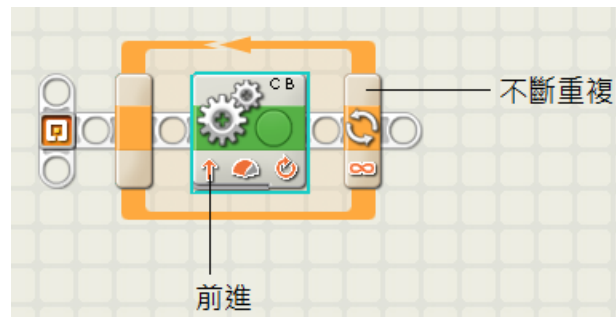


圖 3-2 迴圈

圖 3-2 中橘色的部分為迴圈，代表程式會不斷的執行其內部的指令；而中

間綠色的部分為前進，所以圖 3-2 所代表的意思為「不斷的前進」。

第二個是判斷式，如果「碰撞！」就「轉彎」的概念，在程式中，就是選擇判斷式的概念。NXT Software 中判斷式的應用方式如下：

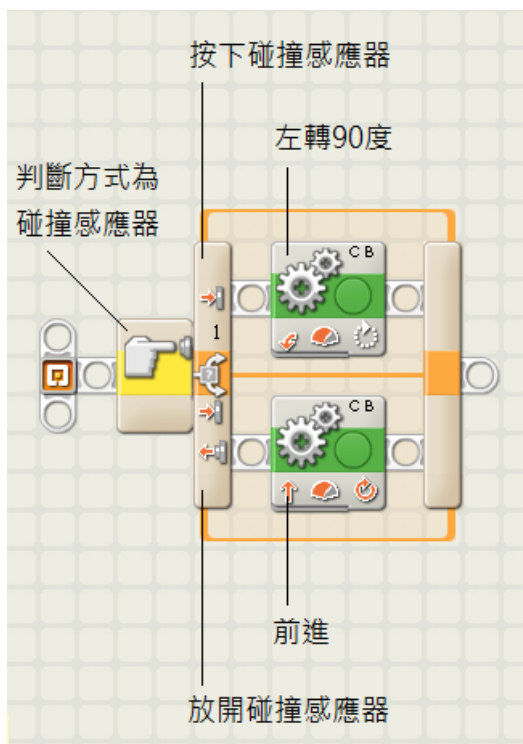


圖 3-3 判斷式

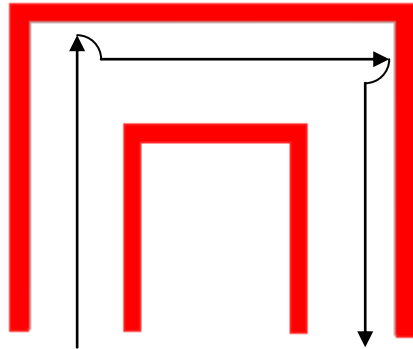
圖 3-3 中黃色的區塊代表此次判斷的方式，而圖中一個手指的形狀所代表的是「碰撞感應器是否被按下」；而橘色則分為上下兩部分，上方代表按下碰撞感應器，其內部綠色區塊所代表之指令為「左轉 90 度」；下方代表放開碰撞感應器，其內部綠色區塊所代表之指令為「前進」。

因此，圖 3-3 整體所代表之指令為「若按下碰撞感應器則左轉 90 度，若放開則前進」。

3-2 自動巡邏！

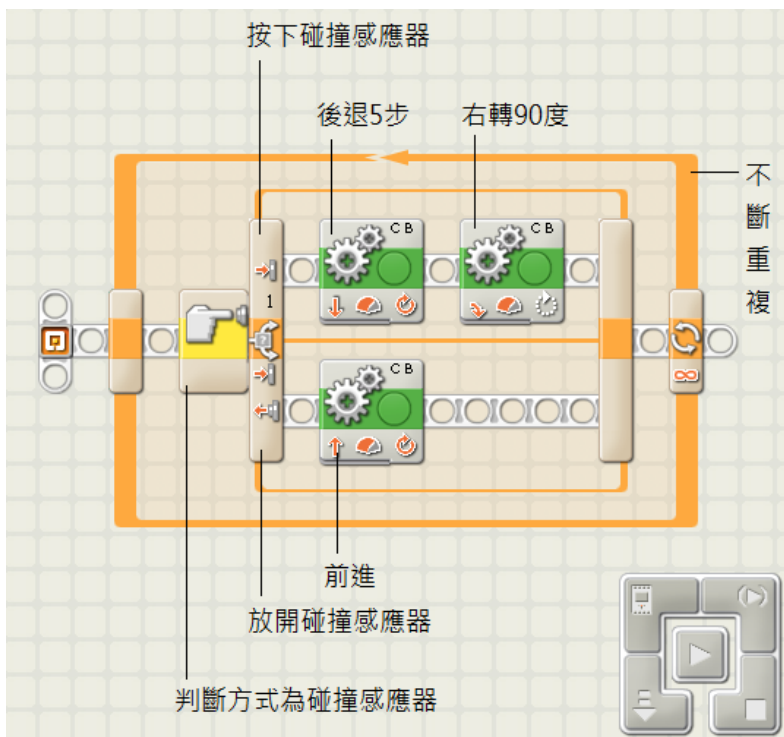
例題：

請設計一程式，使得機器車每當碰到牆壁時就會自動右轉



問題解析：

1. 從 3-1 的觀察當中我們可以看到機器車會一直重覆「前進」、「轉彎」兩種動作。
2. 當機器車碰到牆壁時，才會進行轉彎的動作；反之，當機器車沒有碰到牆壁時，就前進。
3. 若是碰到牆壁就直接轉彎可能會卡在牆邊，所以在碰到牆壁要轉彎之前應該要先後退一些避免卡住。
4. 綜合上述想法，轉化成為指令



3-3 智能巡邏機器車！

習題：

請設計一程式，使得機器車每當靠近牆壁時就會自動右轉。

請問機器車要如何行進才能達到目的地呢？

根據上一題，你認為要用哪些指令區塊組合才能讓機器車達到目的地？請在下方畫出：

提示：距離感應器的使用方法

1.從此拉出判斷式

2.選擇距離感應器

3.調整距離

Control: [Ultrasonic Sensor]

Sensor: [Ultrasonic Sensor]

Port: 1 2 3 4

Compare: [Distance] [50]

Distance: [50]

單元四 光感應器與判斷式進階應用

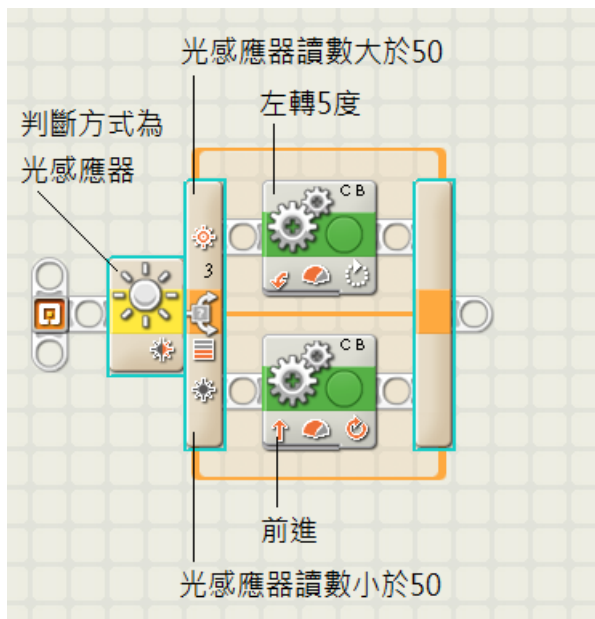
4-1 光感應器

在前一個單元，我們在地圖上有畫圖形，並設計程式讓機器車走同樣的圖形，這種方式必需要知道每一邊的邊長，以及每次轉的角度。假設現在要走的圖形是不規則的或者是圓形，那麼要如何讓機器車沿著軌跡走呢？

本單元要介紹「光感應器」，我們可以將它想像成機器車的眼睛，有了光感應器，機器車可以知道目前的路面的「亮度」，光感應器得到的亮度數值範圍從0~100，數字越大表示亮度越高。光感應器可以傳回一個整數，代表現在機器車所在的路面的亮度。

我們可以配合判斷式及迴圈，將光感應器的指令作為判斷條件，使用的指令如下：

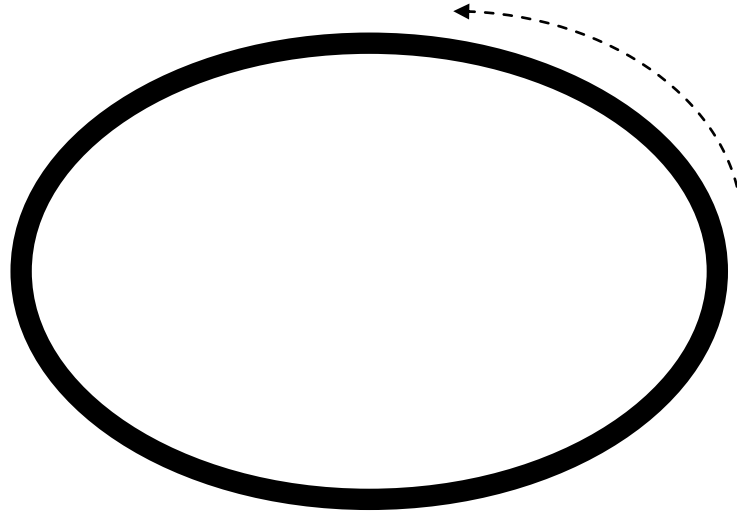
使用光感應器作為判斷條件的例子：



4-2 自動跟蹤機器車！

習題：

請設計一個程式，讓機器車能夠以「逆時鐘方向」沿著橢圓形的黑線行走。



請問機器車要如何行進才能達到目的地呢？

根據上一題，你認為要用哪些指令區塊組合才能讓機器車達到目的地？請在下方畫出：

單元五 樂高機器車綜合應用

5-1 重要觀念介紹

在有了光感應器之後，機器車可以沿著黑線行走，上一個單元，我們採用的策略是利用黑白的分界點來決定車子的前進和轉彎這兩種動作。如果今天我們把圖形改成圖 5-1 有左有右彎的圖形，雖然一樣可以用分界點的策略，若機器車走到黑色右彎，白色左彎，也可以走出這個圖形，但是會走得很不順，或者是原地轉過來轉過去。為了讓機器車可以走得更順暢，我們將地圖的黑線稍微調整了一些，如圖 5-2，調整成漸層的線。而機器車沿線走的策略也從一個分界點，調整成兩個分界點。

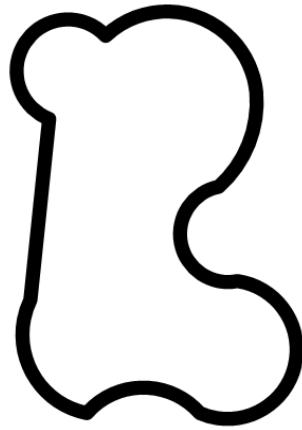


圖 5-1 有左彎有右彎的圖型



圖 5-2 有左彎有右彎的圖型（漸層）

我們先觀察一下漸層線，如圖 5-3，它的光感應值從最黑到白色是漸增的，沒有明顯的分界點，而之前的純黑和純白有明顯的分界點。漸層線的好處就在於此，可以讓我們自由訂定分界點，在此我們以兩個分界點來將漸層線區分為三個部份：較黑、較白、中間；讓車子可以做依不同的區域做出左轉、右轉和前進三種動作。

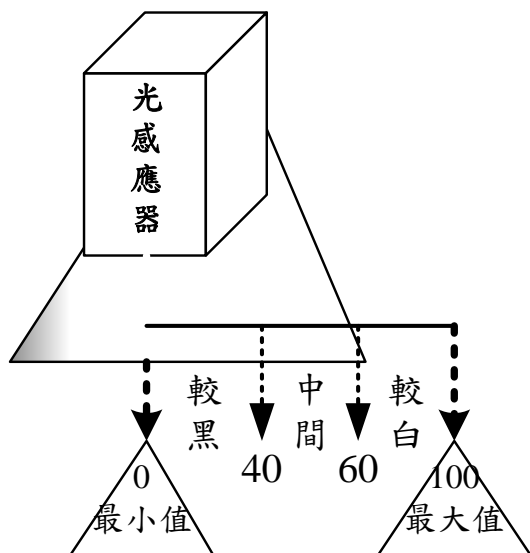
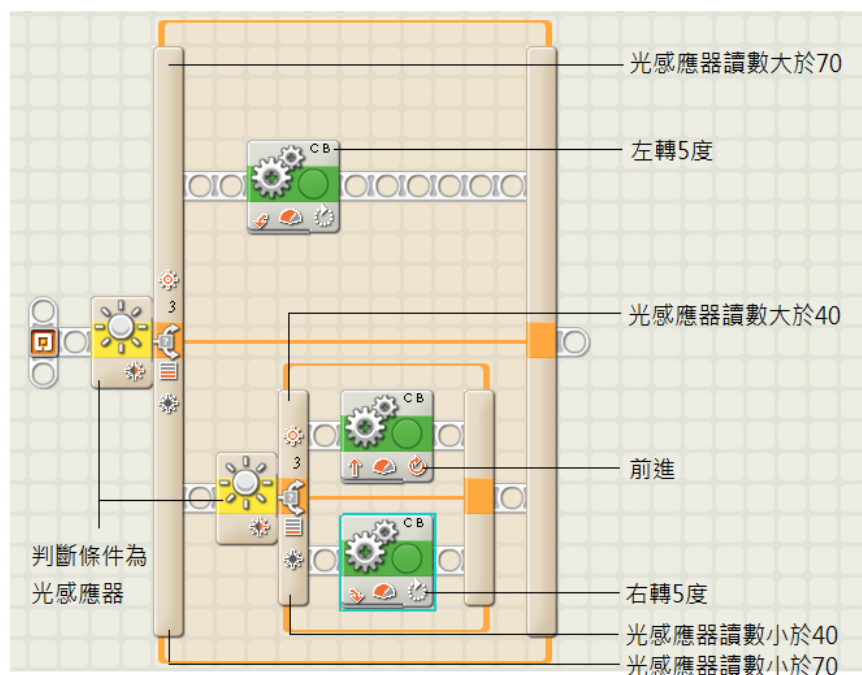


圖 5-3 光感應器所偵測到的亮度值

5-2 多重條件控制的判斷式

在第三單元我們學到了判斷式，現在要講解的是判斷式的一種應用方式「多重條件控制」，顧名思義它可以有多個控制條件，然後程式依序判斷是否有符合的條件執行。下圖為 NXT Software 中「多重條件控制」之指令：



5-3 例題與解析

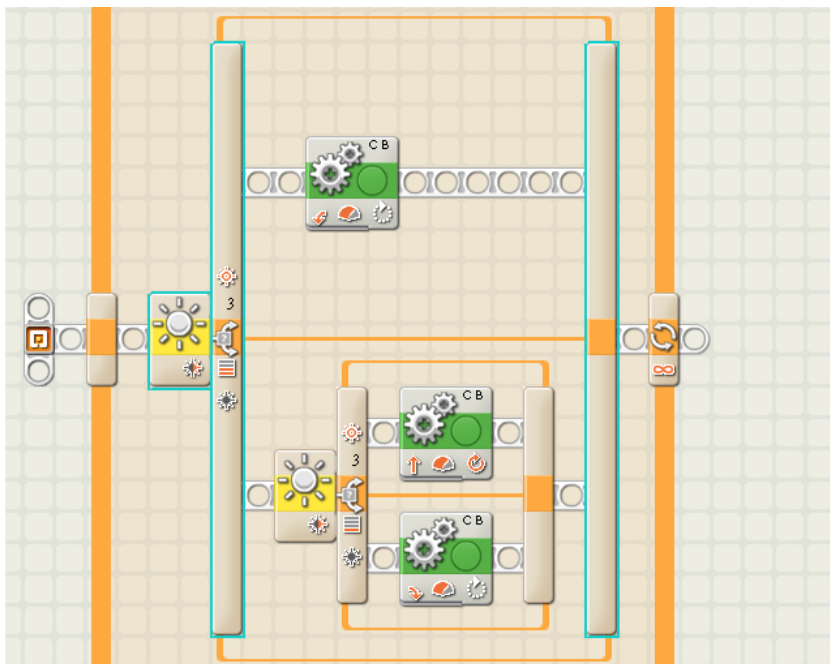
例題：

請設計一個程式，讓機器車可以沿著圖形走一圈。



問題解析：

1. 因為圖形有左彎有右彎，所以機器車必須要會有左彎、右彎、直走三種行為。
2. 從圖 5-3 可以看到光感應的值是從 0 漸漸增加到 100，所以可以將它切割成：
(a) 較黑 (40 以下) (b) 中間 (40~60) (c) 較白 (60 以上)，三個部份。
3. 有三個部份要判斷，所以會用到 5-2 所教的多重判斷式。
4. 車子要逆時鐘走一圈，所以碰到較黑的部份 (a) 要向右彎，碰到較白的部份 (b) 要向左彎，中間的部份就前進。將想法轉換為程式碼如下：

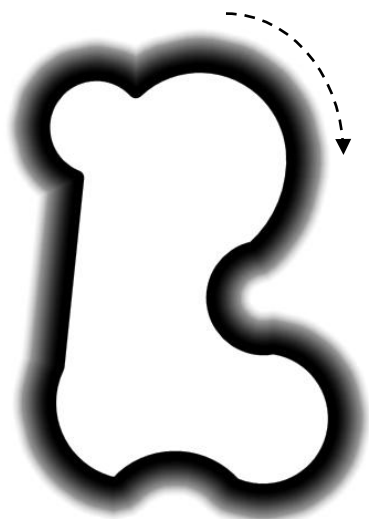


5.

5-4 進階跟蹤！

習題：

請設計一個程式，讓機器車可以沿著圖形走，當碰到障礙物時就停止。



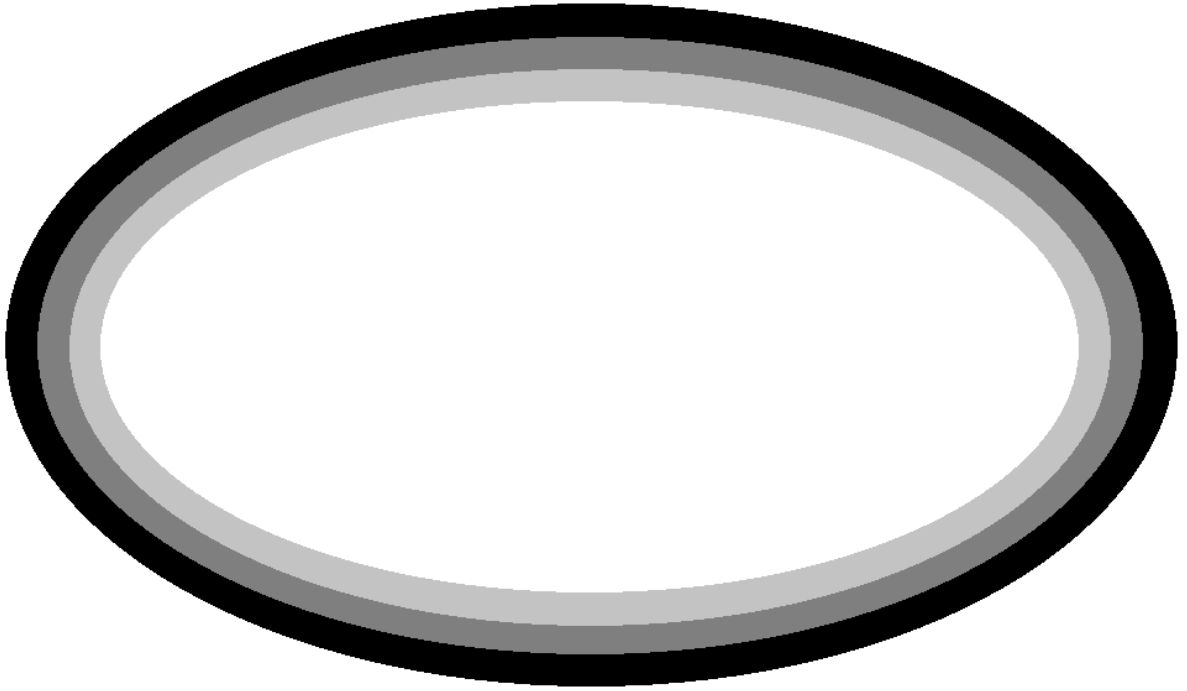
請問機器車要如何行進才能達到目的地呢？

根據上一題，你認為要用哪些指令區塊組合才能讓機器車達到目的地？請在下方畫出：

機器車相撲大賽!

各小組撰寫程式，讓機器車在下圖圓圈中，與對手戰鬥，規則如下：

1. 機器車先停止就輸了。
2. 機器車超出黑線就輸了。
3. 機器車翻車就輸了。



附錄 B

機器車程式設計學習活動問卷

為了瞭解本學期程式設計教學的實施情形，以做為未來教學的改善，請您忠實回答以下問題。

1. 我的性別：男 女

2. 請各位同學用「V」勾選下列題目的同意程度：

問 題	非常 同意	同 意	普 通	不 同 意	非常 不同 意
1. 我覺得機器車程式設計很好玩。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. 我覺得機器車程式設計很容易理解。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. 我覺得機器車程式設計課程很有趣。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. 我能夠了解機器車程式的指令。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. 我可以由閱讀機汽車程式，想像出機器車可能的行動。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. 給我題目，我就可以自己想出解題步驟(不需要老師提示)。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. 給我題目，我不只可以自己想出解題步驟，還能自己組合出程式指令。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. 若機器車行動不如預期，我可以自己找出錯誤的地方。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9. 我覺得機器車程式設計並不會很抽象。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. 我覺得機器車程式設計一點也不好玩。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11. 我覺得完成機器車程式很有成就感。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12. 我覺得機器車程式設計很有挑戰性。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13. 我希望學習更多的程式設計課程。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14. 我覺得學習機器車程式設計很輕鬆。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15. 學習機器車程式設計讓我感到緊張。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16. 我不是很認真的學習機器車程式設計。	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>