

Drape 程式的編修及應用

第一章 緒論

本章簡述國中資訊教育現狀與教授程式設計時所遭遇的問題，進一步說明 Drape 可以作為程式設計入門軟體的原因。由於原 Drape 程式設計者不再開發此軟體且同意研究者修改程式原始碼，故本研究希望能修改原程式不足之處，以發展一套適合國內學生使用的程式語言軟體。

第一節 研究背景與動機

在資訊化的社會中，培養每個國民具備資訊知識與應用能力，已為各國教育發展的重點。政府自 1994 年由行政院規劃國家資訊通信基礎建設計畫，成立國家資訊基礎建設推動小組，以提升我國整體競爭力，教育部也自 1993 年至 1997 年推行補助國中、小設置電腦教室計畫，為學校資訊教育之硬體建設奠基。近來配合課程改革亦將資訊教育列為九年一貫課程之重大議題，根據資訊教育課程(<http://www.ftes.tyc.edu.tw/~ablee/index.html>)，教育部於 2000 年 9 月公佈資訊教育課程綱要，明訂資訊教育課程目標『旨在培養學生資訊擷取、應用與分析、創造思考、問題解決、溝通合作的能力以及終身學習的態度。』，資訊教育課程因此納入國中課程內容。研究者在國中任教，就在那時開始教授資訊課程。

研究者自任教資訊課程以來，即不斷地以九年一貫課程能力指標中所揭示的學習內涵為根本來蒐集補充資料。根據九年一貫課程 (<http://teach.eje.edu.tw/9CC/discuss/discuss2.php>)七到九年級資訊教育學習內涵，中學生需學習的內容包括：電腦的架構、多媒體電腦、程式語言、圖表製作、簡報軟體、問題解決與規劃、認識網路犯罪、正確使用網路的態度以及善用網路科技擴大人文關懷資訊教育議題。對於上述的學習內涵，研究者發現，程式語言是最艱難的學習內容，不只老師教授較少，學生也不感興趣，相對的，學習資料也最少。然而程式語言卻是了解電腦科學的重要知識與技能，所以研究者對學習程式語言的相關軟體就特別留意。

在尋找學習程式語言軟體的過程中，研究者在網路上發現了 Drape(DRAWing Programming Environment 繪圖程式環境)這套軟體(<http://www.cs.uu.nl/~markov/kids/Drape.html>)。Drape 是 Overmars 為了教導孩童程式設計的基本概念，參考 LOGO 繪圖軟體的功能而設計出的視覺導向(visually oriented)程式。當研究者試用此軟體時，發現它非常地易學易用，不需要用鍵盤鍵入程式指令，而用小圖示(icon)代表程式指令，只要將小圖示拖曳到適當位置即可執行繪圖程式，這樣即使完全不懂指令，也能馬上操作。尤其難能可貴的是，它可以在程式執行時同步顯示程式執行的步驟，如此不但對程式除錯

大有助益，也讓學習者對於程式的運作流程及演算法一目了然。研究者認為，Drape 應該可以使國中學生了解程式語言，並了解其在解決問題上的應用。不過研究者更進一步探究後發現，Drape 只能處理一些簡單的數學運算，中學生會用到的開根號或三角函數等功能付之闕如，並且它的介面是英文的，可能學生難以接受。由於 Drape 程式已經不再開發，經與原程式設計者聯繫，原程式設計者同意研究者修改並可逕行散佈新版本 Drape 程式，故研究者擬由教學與研究中嘗試編修 Drape 程式。

第二節 研究目的

基於上述的研究背景與動機，本研究旨在探討 Drape 程式在國中電腦教學上的可行性，研究並延伸 Drape 程式應用的可能性，作為學生學習程式語言以及資訊教育工作者的參考。本研究具體的目的如下：

- 一、探討 Drape 程式在國中施教的可行性。
- 二、Drape 程式中文化。
- 三、延伸 Drape 程式基本功能的探究。
- 四、以 Drape 程式開發解決問題之應用程式範例。

第三節 名詞解釋

一、Drape 程式簡介：

Drape 程式是由荷蘭烏翠特大學(Utrecht University)資訊電腦科學系教授 Mark Overmars 所設計。Overmars 曾經設計過一些給小孩子操作的免費軟體，如 Motion game(動作遊戲)、Tangram(七巧板)、Drawing for children(兒童繪圖)等，而 Drape 程式是教導小孩子一些基本的程式設計的觀念，它是參照 LOGO 程式(中譯：中文小海龜)而寫出的圖形化介面的程式，Drape 程式(<http://www.cs.uu.nl/~markov/kids/drape.html>)目前最新的版本是 2.0 版(Jan 22,1999)，有英文、德文、義大利文以及波蘭文的翻譯說明。

二、Drape 程式語言：

程式語言就是指揮電腦來為我們處理問題並能依順序來執行的指令集(李肇魁、林合彥，1999)，而 Drape 滿足上述對程式語言的定義。Drape 程式的結構，是以小圖示當作指令，拖曳小圖示至適當位置即可執行。程式中的小圖示不多，卻包含程式語言最精要的部分，經由指令圖示的組合可以呈現迴圈(Loop)、變數設定(Set Variable)、呼叫程序(Call Procedure)、If 陳述句(If Statement)等概念。因為輸出指令設定在圖形介面中，不需如傳統程式語言使用鍵盤輸入指令，故 Drape 程式為一視覺導向(visually oriented)的程式語言。

三、LOGO 繪圖軟體：

LOGO 是 1967 年由麻省理工學院(M.I.T)人工智慧實驗室與柏特公司(Bolt,Beranek & Newman Inc.)進行專案研究所發展的電腦語言，從那之後在全球各地實驗室中發展了許多不同的版本。本研究所指的 LOGO 語言，若未特別指定則概括包含立威出版社出版的『小海龜中文 LOGO 語言』，雲林科技大學電子與資訊研究所發展的 CLOGO10 以及由國內程式設計者劉敬洲所寫的『葛拉堡中文小海龜』2.62 版。

四、演算法：

演算法是指「為解決某一問題的一系列有次序且明確的步驟」，好的演算法必須滿足下列五個條件(李肇魁、林合彥，1999)：

- 1.有限性：要在有限的步驟內解決問題。
- 2.明確性：每個步驟都必須清楚地表達出來。
- 3.輸入資料：包含零個或一個以上的輸入資料。
- 4.輸出資料：至少產生一個輸出。
- 5.有效性：必須在有限的時間之內完成。

第四節 研究方法

本研究先將 Drape 與 LOGO 程式作一比較，分析其中的差異，作為程式修改的依據(於第二章說明)。接著將原程式導入教學並進程式修改，藉由教學反省及與 LOGO 程式比較，來修改程式介面以符合實際教學需求(於第三章說明)，最後開發應用程式並探討其延伸功能，以及新增功能以解決問題提供未來研究及教師教學參考(於第四章說明)。

第二章 文獻探討

本章簡述 LOGO 程式語言與其在教育上的應用研究，研究者藉此了解 LOGO 程式在國內教授的情形，來發展 Drape 課程內容。接著對 LOGO 與 Drape 的操作、指令與內建的數學函數作比較，當作 Drape 程式在第一次編修時的參考。最後研究者就自身操作的經驗，概述 Drape 程式的特性。

第一節 LOGO 程式語言與其在教育上的應用研究

LOGO 程式語言是在 1960 年代由美國麻省理工學院 (MIT) 人工智慧實驗室和柏特公司(Bolt,Beranek&Newman Inc.)進行專案研究所發展的電腦語言。LOGO 的目的在提供使用者一個良好的思考情境，使使用者能將它當成一種思考的對象。

Taylor (1980) 在 “The computer in the school ” 一書中指出電腦除了扮演教學者 (Tutor) 的角色外，還有解決統計分析、文書處理、資料管理等問題的工具 (Tool) 角色，以及接受指令、執行動作的「受教者」(Tutee) 角色。LOGO 程式便是具備「受教者」(Tutee) 角色的人工智慧電腦輔助學習軟體。再者，它有明顯易懂的指令以及立即的視覺回應，適合電腦初學者學習程式設計，故被國內外學者專家推崇為特別適合中小學生學習的一種程式語言。

LOGO在國內較早的中文版本，是立威出版社出版的『小海龜中文LOGO語言』(1994)，1999年雲林科技大學人工智慧實驗室研發的『網路版中文 CLOGO』，是一個以網頁互動為主的海龜繪圖介面系統，而目前較為普及的版本，是由劉敬洲所寫的「葛拉堡中文小海龜」，為各學者研究時所採用(鍾仁貴，1999；黃文聖，2000；林裕雲，2001)。

相較於國外的研究，國內對LOGO相關的研究並不多(黃文聖，2000)。曾錦達(1995)曾探討國小六年級學童實施LOGO 程式設計(繪圖)教學的可行性，並分析學童學習LOGO 時常犯的錯誤。從他的研究分析中獲致的結果如下：國小實施LOGO 教學的可行性有必要再做進一步的評估；學習LOGO 程式設計對國小學童程序性思考能力的效應雖尚未達顯著水準，但從統計資料中顯示，仍有一定程度的正向效果。

崔夢萍(1999)研究國小五年級學童共 117 人，依照班級分為Logo實驗組、多媒體實驗組、及控制組進行六週之教學。採用「陶倫斯創造思考圖形評量圖形測驗」來評量學童之創造力，研究結果發現多媒體組比控制組和LOGO組，顯著地增加其在獨創力、流暢力、變通力、及精進力方面的創造思考力；LOGO 組和控制組的學童在各方面都沒有差異，只有在精進力上，Logo實驗組的學生比控制組的學生創造思考力顯著地降低。

黃文聖(2000)認為LOGO 的學習環境可以幫助學童主動、獨立地思考，並能將抽象的數學概念具體化，讓學童得以提早學習較複雜的數學知識。另外，林裕雲(2001)也研究發現學生接受「LOGO 課程」會導致其解題能力的進步。

研究者就現有研究搜尋結果發現，學者對 LOGO 的教學大致上持肯定態度，然而對

程式是否增進學童的創造與思考能力有不同的結論。可能原因為一方面學生創造思考過程太過於複雜，另一方面研究的深度及廣度不足，故無法達到顯著效果。

另一方面，因 LOGO 程式在數值的運算表示方式不比 Basic 程式來得易讀易寫，故教材上大多省略不提數值運算的部分。LOGO 的長處是在於繪圖，目前 LOGO 使用於教學上以繪圖為主，藉由畫出一些規則或對稱圖形來了解程式及程序，對象由小學至高職生，凡電腦初學者皆相當適合。然而畫出圖形與數學幾何圖形類似，以致學生缺乏學習動機，是它的隱憂(崔夢萍，1999)。

使用 LOGO 時，學習者以鍵盤輸入幾個簡單的英文字如 FORWARD、RIGHT、LEFT、REPEAT、HOME 等，就可繪製簡單圖形，例如打入『REPEAT 4 [FORWARD 100 RIGHT 90]』的指令，即可畫出一正方形。若為了方便輸入而使用簡碼時，上面的例子可寫成『RP 4 [FD 100 RT 90]』，縮短了程式的長度。LOGO 中文化之後操作更加簡單，以劉敬洲所寫的「葛拉堡中文小海龜」而言，作者將繪圖最常用到的 16 個指令連同數字 1 到 10 作成按鍵，只要用滑鼠按指令再按數字即可顯示程式並執行，例如『重複 4 [前進 100 右轉 90]』的指令，除了「[」與「]」是鍵盤輸入外，其餘皆可以滑鼠點選。

研究者約於 1999 年開始學習 LOGO 語言，認為它是眾所公認，容易學習的繪圖軟體，但在寫程式上，研究者認為 LOGO 語言不如 BASIC 程式那樣簡明，尤其在 LOGO 語言中的遞迴流程，相當難以了解。當研究者使用 Drape 程式之後，發現 Drape 揉合了 LOGO 繪圖的直觀性與 BASIC 程式的簡潔性，再加上視覺導向的程式設計，應該有助於學生掌握繪圖與程式設計的觀念，因此選擇 Drape 程式當研究對象。

第二節 Drape 程式的操作特性

使用 Drape 的學習者不需使用鍵盤就可繪製簡單的幾何圖形，只要拖曳適當的指令並執行即可，原作者細心地將最常用的指令製成小圖示，放在簡單(easy)以及正常(normal)標籤頁中。以簡單(easy)標籤頁為例，常用的指令有向前移動一步、向後移動一步、向前畫線一步、向後畫線一步、放大、縮小、左轉 90 度與 45 度、右轉 90 度與 45 度、重複 2、3、4、5、10、100 次。僅僅使用上述指令，利用加成性的關係就可以組合出許許多多的幾何圖形。

在 Drape 中的加成性有兩種：加法與乘法。很顯然地，長度是加法：向前移動一步，再向前移動一步，就等於向前移動兩步，將這些移動指令(較小的一步)放在一起可以得到較大的一步。同樣，角度也是加法，將右轉 90 度以及右轉 45 度的指令放在一起，游標就向右轉了 135 度。另一方面，放大與縮小是乘法的，將放大 2 倍後面加一個放大 2 倍，可以得到放大 4 倍的結果(放大與縮小只是使移動的單位長度放大或縮小，不影響畫線寬度與角度)。同樣地，重複也是乘法，重複 2 次指令加上重複 3 次指令可以得到重複

6 次的結果。

要更有彈性的輸入，可以到正常(normal)標籤頁中使用諸如旋轉 6 度或是放大(縮小)一點點的指令，在簡單 (easy)與正常(normal)標籤頁中，拖曳指令並排列組合就可以創造無數新圖形，這些是完全不需鍵入一行程式碼就可以做到的事，然而，增加方便性意味著犧牲了精確性，比方說僅用拖曳圖示永遠無法達到像是前進 1.3 單位長，旋轉 22.5 度，放大 1.1 倍或是重複 7 次的功能。Drape 提供進階 (advanced)標籤頁指令，此標籤頁中的指令可以自行鍵入數值或數學函數，而這時就需要鍵盤輸入了，所以 Drape 的學習門檻很低，同時也可以做精確的運算。




第三節 LOGO 與 Drape 的指令比較

茲以立威出版社出版的『小海龜中文LOGO語言』與尚未修定的Drape程式作指令比較，指令功能、LOGO碼與簡碼引用自小海龜電腦繪圖—中文LOGO語言實務手冊(p207—p208)。表 2.1 列出兩者皆有的指令，表 2.2 列出Drape獨有的指令，表 2.3 列出LOGO獨有的指令，作一比較如下：

表 2.1 LOGO 與 Drape 皆有的指令

編號	指令功能	LOGO 碼	簡碼	Drape	說明
1	前進	FORWARD	FD	 或 	
2	後退	BACKWARD	BK	 或 	
3	右轉	RIGHT	RT		
4	左轉	LEFT	LT		
5	提筆	PENUP	PU	 或 	
6	下筆	PENDOWN	PD	 或 	
7	隱藏游標	HIDETURTLE	HT	 off	
8	出現游標	SHOWTURTLE	ST	 on	
9	回家	HOME	HM		
10	清圖	CLEARSCREEN	CS		
11	印出	PO	PO	總是顯示	顯示某程序內容
12	印全部	POALL	POALL	總是顯示	顯示所有程序內容
13	印頭	POTS	POTS	總是顯示	顯示目前程序名稱
14	到	TO	TO	6 個程序	
15	結束	END	END	指令結束即停止	

16	目錄	DIRECTORY	DIR		查看磁碟內容
17	編輯	EDIT	ED	總是可以編輯	編輯一個已經存在的程序
18	擦掉全部	ERASEALL	ERALL		刪除所有程序
19	擦掉	ERASE	ER	滑鼠拖至垃圾桶	刪除程序
20	存檔	SAVE	SAVE		
21	取檔	LOAD	LOAD		
22	存圖	SAVEPIC	SAVE PIC		
23	取圖	LOADPIC	LOAD PIC		
24	離開程式	EXIT	EXIT		
25	重複	REPEAT	RP		
26	變數	MAKE	MK		設定數值變數
27	如果	IF	IF		
28	否則	ELSE	ELSE		
29	區塊	[]	[]		
30	停止	STOP	STOP		
31	定位	SETPOS	SETPOS		
32	角度	SETH	SETH		
33	塗色	FILL	FILL		
34	變色	COLOR	COLOR		
35	迴圈	WHILE	WHILE		
36	全圖	FULLSCREEN	FS	 on	全圖模式
37	全文	TEXTSCREEN	TEXT		程式碼模式
38	繪圖	DRAWSCREEN	DRAW	 off	圖文並存模式
39	筆色	SETPENCOLOR	SETPC		
40	字色	SETTEXT COLOR	SETTC		
41	底色	SETBACK COLOR	SETBK		
42	清除	CLEAN	CN		清除螢幕

43	換錄	CHDIR	CD		
44	印字	PUTTEXT	PT		
45	版本	VERSION	VER		

上表列出 45 項 LOGO 與 Drape 皆有的指令，大體上它們的主要繪圖功能都差不多，不過 LOGO 是以輸入英文字執行程式，而 Drape 則是以拖曳小圖示執行程式，以小圖示代替文字敘述，至少有下列三項優點：

- 一、程式較為直觀：不是所有人都懂 LOGO 的英文指令，但 Drape 的小圖示人人都看得懂，並且能在很短的時間內學會操作的方法。
- 二、程式大幅縮短：小圖示比文字指令甚至是簡碼佔更少的空間，大幅縮短程式長度。LOGO 是由文字指令建構的，寫出來的程式與程序通常很長，以至於無法在一個視窗畫面下顯示，所以需要如編號 11、12、13 的 LOGO 指令，去特別指定顯示某一程序的內容；Drape 是由小圖示指令建構的，寫出來的程式與程序通常可以在一個視窗畫面下全部顯示，因此，它可以顯示全部的程式內容。對 LOGO 而言，由於無法同時顯示所有程序，所以每個程序都要命名，以備呼叫或編輯(編號 17，EDIT 指令)之用；對 Drape 而言，不需對程序命名，有六個鑽石圖示來放這些程序(說明見第三章第五節)，在不執行程式時就是編輯狀態，不用特別下編號 17 的編輯指令。
- 三、指令可以複合：就以顏色設定為例，在 LOGO 中有變色、筆色、字色、底色(編號 34、39、40、41)四種指令，Drape 使用一種顏色設定指令，搭配其他指令複合而成也可以達到相同目的，一來增加程式的可讀性，創造了不同的意義，同時也簡化了指令的數目。

表 2.2 Drape 獨有的指令

編號	指令功能	LOGO 碼	簡碼	Drape	說明
1	慢速執行	無	無		
2	一次執行一指令	無	無		
3	快速執行	無	無		
4	暫停程序	無	無		
5	畫正方形	無	無		
6	畫長方形	無	無		
7	畫扁長方形	無	無		
8	畫實心圓	無	無		

9	畫三角形	無	無		
10	畫扁三角形	無	無		
11	放大	無	無		
12	縮小	無	無		
13	移到任意位置	無	無		
14	畫到任意位置	無	無		
15	移到滑鼠位置	無	無		
16	畫到滑鼠位置	無	無		
17	按左鍵時執行下一步	無	無		
18	按右鍵時執行下一步	無	無		
19	不按鍵時執行下一步	無	無		
20	偵測底色	無	無		
21	聲音播放	無	無		
22	休息	無	無		
23	按下某鍵執行下一步	無	無		
24	環境切換	無	無	 	環境變數的儲存與設定
25	執行中開啓檔案	無	無		

有 25 項功能是 Drape 有，而 LOGO 沒有的，其中編號 1 到 4 與程式執行有關，Drape 可以任意地調整程式的執行速度，這是 LOGO 無法做到的。編號 5 到 10 是 Drape 可以畫出某些特定的幾何形狀。編號 11、12 是 Drape 允許用一種放大/縮小的觀念來定義單位長度。編號 13，14 引進亂數的概念。編號 15 到 19 引進滑鼠控制的功能。編號 20 偵測底色。編號 21 可匯入聲音檔案。編號 22 讓程式執行得慢一點。編號 23 按鍵控制。編號 24 為環境變數的儲存與設定。編號 25 是執行中開啓其他檔案。這些功能都是 Drape 所獨有的。

表 2.3 LOGO 獨有的指令

編號	指令功能	LOGO 碼	簡碼	Drape	說明
1	印字串或變數	PRINT	?	無	
2	字串	MAKESTR	MKS	無	設定字串變數
3	顯示記憶體	SPACE	SPACE	無	

	空間				
4	印圖	PRINTGRAPH	PRTGR	無	印出圖形至印表機
5	印幕	PRINTSCREEN	PRT SCR	無	印出螢幕至印表機
6	顯示模式	DISPLAYMODE	DM	無	設定中文/英文命令顯示

與 LOGO 程式相較下，原 Drape 程式還是有不足之處，由編號 1 及 2 可以看出，由於 Drape 無法顯示變數的功能，也不能處理字串變數，這兩項功能是 Drape 可以考慮增加的功能。至於編號 3、4、5 的功能可以用其他方式達成，編號 6 的功能在程式中文化之後應該是不需要的。研究者考量後認為，字串變數的使用機會並不多，在修改 Drape 程式時暫不增加這種功能；但不能顯示變數限制了程式中數字處理的部份，所以若是可能，應該要在編修 Drape 程式中加入此項功能。

第四節 LOGO 與 Drape 內建數學函數比較

相對於 LOGO 語言，Drape 的內建數學函數不多。研究者就葛拉堡中文小海龜、CLOGO10 與 Drape 內建數學函數比較如表 2.4：

表 2.4 小海龜、CLOGO10 與 Drape 內建數學函數比較表

編號	說明	小海龜 2.62 版	CLOGO10	Drape	是否增加	原因
1	加	+	SUM	+	否	已有
2	減	-	DIFFERENCE	-	否	已有
3	乘	*	PRODUCT	*	否	已有
4	除	/	QUOTIENT	/	否	已有
5	絕對值	ABS	ABS		否	已有
6	橫座標	XCOR	XCOR	x	否	已有
7	縱座標	YCOR	YCOR	y	否	已有
8	圓周率	PI		pi	否	已有
9	平方根	SQR	SQRT	無	否	使用次方功能即可
10	亂數	RND	RANDOM	random	否	已有
11	對數	LOG		無	否	國中階段用不到
12	正弦	SIN	SIN	無	是	
13	餘弦	COS	COS	無	是	
14	自然對數	EXP	LN	無	否	國中階段用不到

15	次方	^	POWER	無	是	
16	正切	無	TAN	無	否	使用正弦，餘弦即可
17	反餘弦	無	ARCCOS	無	否	使用反正切即可
18	反正弦	無	ARCSIN	無	否	使用反正切即可
19	反正切	無	ARCTAN	無	是	

由上表可知 Drape 只有簡單的數學運算，與 LOGO 語言比較起來沒有次方功能、三角函數、反三角函數以及對數、指數功能，研究者擬將較常用的次方功能、三角函數、反三角函數功能建置於 Drape 中，以延伸 Drape 應用的範圍。

第五節 Drape 的特點

在試用過 Drape 程式以及與 LOGO 程式比較後，研究者認為 Drape 有以下特點：

- 一、視覺導向的操作環境：Drape 將程式碼指令轉成圖形，除了方便撰寫程式，也使使用者操作錯誤的次數減到最低。程式設計初學者以繪圖功能入門，不僅學習門檻低，程式執行結果也常讓人驚異，進而激發學習動機。
- 二、指令呈現的群組化：一般軟體會適度隱藏指令，免得指令呈現過多時令人眼花撩亂，Drape 更進一步，將指令群隱藏於『簡單』、『正常』、『進階』三個標籤頁中，每個指令群的指令數不多，然而足以獨立執行程式功能而不須切換標籤頁，三個標籤頁中的指令並非毫無關聯，而是互相聯繫，循序漸進的，提供了初學者、進階者不同的使用介面。
- 三、基本的程式語句：Drape 提供的指令，在傳統程式語言表示法中，有迴圈(loop)、變數設定(Set Variable)、呼叫程序(Call Procedure)、If 陳述句(If Statement)、環境變數、檔案存取等概念。在視窗程式中有游標下的顏色指令、滑鼠指令、按鍵指令等，可以做不同組合而產生新意義，包含程式設計基本概念，也適合初學程式設計者使用。
- 四、豐富的數學意涵：從程式介面來看，畫圖區即是一個絕對直角座標系統，或是無限多個相對直角座標系統可管轄的範圍，使用者可以用直角座標的觀念、極座標的觀念或參數式的觀念來畫圖。游標前進與後退可視為加法、減法，重複區塊指令可視為乘法，Drape 可以使用絕對值、亂數、and 以及 or 邏輯判斷等，若再加上三角函數、乘幂等函數後應該可以用來解決一些數學、數值問題。
- 五、流暢的操作流程：相較於其他程式語言軟體，Drape 不須編譯，直接在執行檔下執行即可，它的執行檔大小不到 1MB，Drape 對軟硬體要求不高，即使在 Windows95 下執行也相當穩定不當機。使用慢速執行或一步一步執行讓使用者看到程式執行流程，方便使用者除錯。Drape 對錯誤的包容力強，例如當程式碼中只有開始區塊，

無結束區塊，則程式依舊可以執行。錯誤訊息只有在重大瑕疵下，如變數計算找不到變數名稱或輸入的數學算式有問題才會顯現。

綜合上述，**Drape** 是容易學習的軟體，適合初學電腦者學習，然而，若能增加一些新功能，將可以拓展程式的應用範圍，因此研究者擬增加數學函數與顯示變數功能，來增強程式功能。





第三章 Drape 的修改

本章概述 Drape 程式修改，內容包括：新增數學函數、顯示變數與精度功能以及程式中文化。第五節詳列 Drape 程式修改之後的操作說明，用 21 個範例程式來說明每一個圖式指令的用法以及程式執行之後的結果。

第一節 新增數學函數

數學函數實作：對照葛拉堡中文小海龜、CLOGO10 與 Drape 內建數學函數比較表以及研究者撰寫程式之經驗，將數學函數 \cos 、 \sin 、 \arctan 、 power 加入 Drape 中，之後研究者發現無條件捨去的功能算餘式很方便，所以再加上小數點下一位無條件捨去的功能。因顧及輸入之方便性，沿襲程式原始碼中以一個字元代表一種數學運算(如『+』、『-』、『*』、『/』)，研究者以『@』代表 \cos 函數、『\$』代表 \sin 函數、『!』代表 \arctan 函數、『^』代表冪次方以及『~』代表無條件捨去，比如說， $@(60)=0.5$ ， $2^3=8$ ， $\sim(1.9)=1$ 等。

第二節 新增顯示變數功能

原 Drape 程式有變數處理的圖示方法，卻沒有變數顯示的圖示方法，因此可以用變數比較來控制程式流程，但無法顯示程式處理之後的變數值。若有顯示變數圖示方法將使 Drape 的功能大大提升，可幫助學習者掌握變數、解決數的問題。研究者修改程式原始碼、加入顯示變數圖示方法，但爲了要盡量維持程式介面的單純，不要增加圖示方法的數量，因此希望找出原 Drape 介面中多餘的圖示，進而取代之。研究者發現，原程式中的 Text 小圖示  是可被替代的，因爲此功能爲 Rotated Text  的一個特例(旋轉角度等於零)。因此取消 Text 小圖示並加入顯示變數  的功能。研究者針對顯示變數，設計新的對話框如圖 3.1。

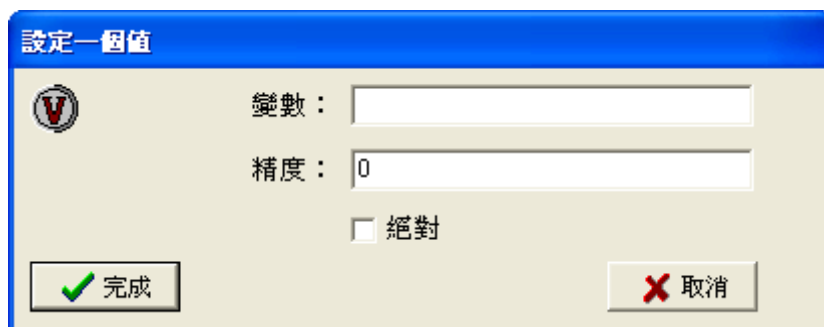


圖 3.1 顯示變數對話框

圖中需輸入變數名稱，以顯示變數，精度代表顯示到小數點下第幾位(預設爲 0，將顯示四捨五入後的整數)，預設絕對不打勾，將顯示變數名稱及數字，若絕對打勾，將只顯示數字。例如，將變數 a 設定爲 $2^{0.5}$ ，輸入顯示變數 a ，精度 3，絕對不打勾，呈

現的結果將是 $a=1.414$ 。若絕對打勾，則只顯示數值 1.414 而不顯示變數名稱。

第三節 修改變數比較的精度

在增加程式顯示變數功能後，研究者測試程式時發現原 Drape 程式的變數比較只適用於整數。例如，若寫一應用程式如下：



原 Drape(英文版)程式列印如下：

```
PROCEDURE 1
begin procedure
  set variable a to 0.6+0.1
  if a=0.7 then
    begin block
      draw the rotated text "a=0.7"
    stop program
    end block
  draw the rotated text "a<>0.7"
end procedure
```



程式首先將 a 設定為 $0.6+0.1$ ，接下來設定若 $a=0.7$ ，則印出 $a=0.7$ 並停止程式；否則印出 $a<>0.7$ 而結束程式。照程式流程，變數 $a=0.7$ ，應該印出 $a=0.7$ 並停止程式，不過執行程式之後，卻發現程式印出 $a<>0.7$ 。研究者發現，因為原 Drape 程式對於非整數變數的比較太過敏感，研究者曾經測試過，變數的比較在小數點後第 38 位數依然有效。在程式中非整數變數經過數學運算後可能產生誤差，因而造成電腦誤判。如上例中非整數變數的相加： $0.6+0.1$ 與數值 0.7 在電腦中是不一樣的，此差異又為 if 條件句所偵測，故程式認定 $0.6+0.1<>0.7$ 。

原 Drape 程式沒有捨棄數學運算的微小誤差，把原來應該相等的數值視為不相等，將造成程式撰寫時的大問題，研究者考慮後，決定降低程式對數值認定的敏感度：若甲乙兩數差的絕對值小於 10 的負 11 次方，則兩數相等，以免造成不必要的困擾。因此，研究者以此標準修改 Drape 原始碼，重新撰寫此程式如圖：



其程式列印如下：

程序 1

程序開始

將變數 a 設為 0.6+0.1

若變數為 $a=0.7$ 則

區塊開始

畫旋轉文字 "a=0.7"

停止程式

區塊結束

畫旋轉文字 " $a < 0.7$ "

程序結束

執行此程式後，就如同所預料的，將顯示 $a=0.7$ 。





第四節 程式中文化

一、小提示的中文化：一般視窗軟體都有所謂的小提示(hint)，當滑鼠游標移到某圖示上一會兒，即會出現一段文字描述此圖示的功能，Drape 原程式的小提示為英文，將其替換為中文如表 3.1。

表 3.1 小提示的中英對照

圖示	英文	中文	圖示	英文	中文
	New Program	開新程式		Load Program	載入程式
	Save Program	儲存程式		Print Program	列印程式
	Show Cursor	顯示游標		Full Screen	全螢幕
	Run Slow	執行(慢)		Run Fast	執行(快)
	Single Step	單一步驟		Pause	暫停
	Stop	停止		Save Picture to a File	存圖至檔案
	Info	資訊		Quit	離開
	Red	紅		Blue	藍
	Green	綠		Yellow	黃
	White	白		Black	黑
	Much Larger	放大		Much Smaller	縮小
	Larger	放大一點		Smaller	縮小一點
	Little Larger	放大一點點		Little Smaller	縮小一點點

	Thin Lines	細線		Normal Lines	一般線
	Thick Lines	粗線		Move	向前移動
	Draw Line	向前畫線		Move Back	往後移動
	Draw Line Back	向後畫線		Move to Center	移到中心
	Rotate Left 90	左轉 90 度		Rotate Left 45	左轉 45 度
	Rotate Left 6	左轉 6 度		Rotate Right 90	右轉 90 度
	Rotate Right 45	右轉 45 度		Rotate Right 6	右轉 6 度
	Draw Square	畫正方形		Draw circle	畫圓
	Draw Triangle	畫三角形		Draw Flat Triangle	畫扁三角形
	Draw Rectangle	畫長方形		Draw Flat Rectangle	畫扁長方形
	Random Move	移到任意點		Random Line	畫到任意點
	Fill the Area	填滿		Clear Image	清除影像
	Move to Mouse	移到滑鼠游標		Line to Mouse	畫到滑鼠游標
	Repeat 2x	重複 2x		Repeat 3x	重複 3x
	Repeat 4x	重複 4x		Repeat 5	重複 5x
	Repeat 10x	重複 10x		Repeat 100	重複 100x
	If Left Mousebutton	若按下左鍵		If Right Mousebutton	若按下右鍵
	If No Mousebutton	若不按鍵		If Red	若為紅
	If Blue	若為藍		If Green	若為綠
	If Yellow	若為黃		If white	若為白
	If Black	若為黑		Begin Block	區塊開始
	End Block	區塊結束		Stop Program	停止程式
	Save Environment	儲存環境變數		Restore Environment	回復環境變數
	Change Color	改變顏色		Change Size	改變大小
	Change Angle	改變角度		Change Line Width	改變線寬
	Move to a Place	移動到		Line to a Place	畫到

	Load Image from File	從檔案載入圖形		Save Image to File	圖形存入檔案
	無	顯示變數		Rotated Text	印文字
	Play Sound	聲音播放		Execute Program from File	從檔案執行程式
	Check for a Color	顏色檢查		Repeat some Times	重複若干次
	Sleep	休眠		Exit current procedure	離開目前程序
	Check for pressed key	按鍵檢查		Set variable	設定變數
	Compare values	變數比較		Call Procedure 1	呼叫程序 1
	Call Procedure 2	呼叫程序 2		Call Procedure 3	呼叫程序 3
	Call Procedure 4	呼叫程序 4		Call Procedure 5	呼叫程序 5
	Call Procedure 6	呼叫程序 6			




二、列印程式的中文化：Drape 中有一列印程式  的功能，是將被拖曳的圖示方法轉換成文字註解後顯示於螢幕上，方便使用者閱讀程式。每個圖示方法都有一個相對應的文字註解，而原來是以英文顯示文字註解，今將英文翻譯成中文如表 3.2。

表 3.2 列印程式的中英對照

圖示	原列印的英文	翻譯後的中文
	set color to red	顏色設定紅色
	set color to blue	顏色設定藍色
	set color to green	顏色設定綠色
	set color to yellow	顏色設定黃色
	set color to white	顏色設定白色
	set color to black	顏色設定黑色
	set size much larger	尺寸放大
	set size much smaller	尺寸縮小
	set size larger	尺寸放大一點
	set size smaller	尺寸縮小一點
	set size a little larger	尺寸放大一點點

	set size a little smaller	尺寸縮小一點點
	set line width thin	線寬設定細線
	set line width medium	線寬設定中等線
	set line width thick	線寬設定粗線
	move one step	向前移動一步
	draw a line one step	向前畫線一步
	move one step back	向後移動一步
	draw a line one step back	向後畫線一步
	move to the center	移到中心
	rotate left 90 degrees	左轉 90 度
	rotate left 45 degrees	左轉 45 度
	rotate left 6 degrees	左轉 6 度
	rotate right 90 degrees	右轉 90 度
	rotate right 45 degrees	右轉 45 度
	rotate right 6 degrees	右轉 6 度
	draw a filled square	畫正方形
	draw a filled circle	畫圓
	draw a filled triangle	畫三角形
	draw a filled flat triangle	畫扁三角形
	draw a filled rectangle	畫長方形
	draw a filled thin rectangle	畫扁長方形
	move to a random place	移到任意點
	draw a line to a random place	畫到任意點
	flood fill from current position	填滿
	erase the image	清除影像
	move to the mouse position	移到滑鼠游標
	draw a line to the mouse position	畫到滑鼠游標
	repeat 2 times	重複 2 次
	repeat 3 times	重複 3 次
	repeat 4 times	重複 4 次
	repeat 5 times	重複 5 次
	repeat 10 times	重複 10 次

	repeat 100 times	重複 100 次
	if the left mousebutton is pressed then	若按下左鍵則
	if the right mousebutton is pressed then	若按下右鍵則
	if no mousebutton is pressed then	若不按鍵則
	if color at cursor is red then	若游標處為紅則
	if color at cursor is blue then	若游標處為藍則
	if color at cursor is green then	若游標處為綠則
	if color at cursor is yellow then	若游標處為黃則
	if color at cursor is white then	若游標處為白則
	if color at cursor is black then	若游標處為黑則
	begin block	區塊開始
	end block	區塊結束
	stop program	停止程式
	save environment	儲存環境變數
	restore environment	回復環境變數
	set color to	改變顏色為
	set size absolutely to	設定絕對尺寸比率為 XX%(當勾選絕對時)
	change size to	設定相對尺寸比率為 XX%(當不勾選絕對時)
	set angle absolutely to XX degrees	設定游標絕對角度為 XX 度(當勾選絕對時)
	rotate with XX degrees	設定游標相對角度 XX 度(當不勾選絕對時)
	set line width to	設定線寬為
	move absolutely to	移動到絕對座標 X Y :(當勾選絕對時)
	move relative to	移動到相對游標座標 X Y :(當不勾選絕對時)
	draw a line absolutely to	畫線到絕對座標 X Y :(當勾選絕對時)
	draw a line relative to	畫線到相對游標座標 X Y :(當不勾選絕對時)

	draw image XX unscaled	從檔案載入圖形 XX 尺寸不變(當勾選絕對時)
		從檔案載入圖形(當不勾選絕對時)
	save image as	圖形存入檔案
	無	顯示變數 XX 的值到小數點下 YY 位(當勾選絕對時)
	無	顯示變數 XX 的名稱及值到小數點下 YY 位(當不勾選絕對時)
	draw the rotated text unscaled	畫旋轉文字 XX 原尺寸(當勾選絕對時)
	draw the rotated text	畫旋轉文字(當不勾選絕對時)
	play sound file	聲音播放
	call program	從檔案執行程式
	if color at cursor is	若顏色為 XX 則
	repeat XX times	重複 XX 次
	sleep for XX milliseconds	休眠 XX 毫秒
	exit the procedure	離開目前程序
	if last key pressed is XX then	若按下 XX 鍵則
	set variable XX to YY	將變數 XX 設為 YY
	If XX then	若變數為 XX 則
	call procedure 1	呼叫程序 1
	call procedure 2	呼叫程序 2
	call procedure 3	呼叫程序 3
	call procedure 4	呼叫程序 4
	call procedure 5	呼叫程序 5
	call procedure 6	呼叫程序 6

第五節 修改之後的操作說明

一、安裝以及解除安裝：

系統需求個人電腦(Pentium 以上)，系統需求為 Windows95 或 NT4.0(以上)，在安裝後程式所佔磁碟空間不到 1 MB。要安裝 Drape，只要將 Drape.exe 複製到你的電腦即可。Drape 不需另外安裝任何系統檔案到電腦上，而要解除安裝，只要刪除 Drape.exe 即可。

二、Drape 的介面：

程式安裝後，點選 Drape.exe 即可進入程式，顯示畫面如圖 3.2：

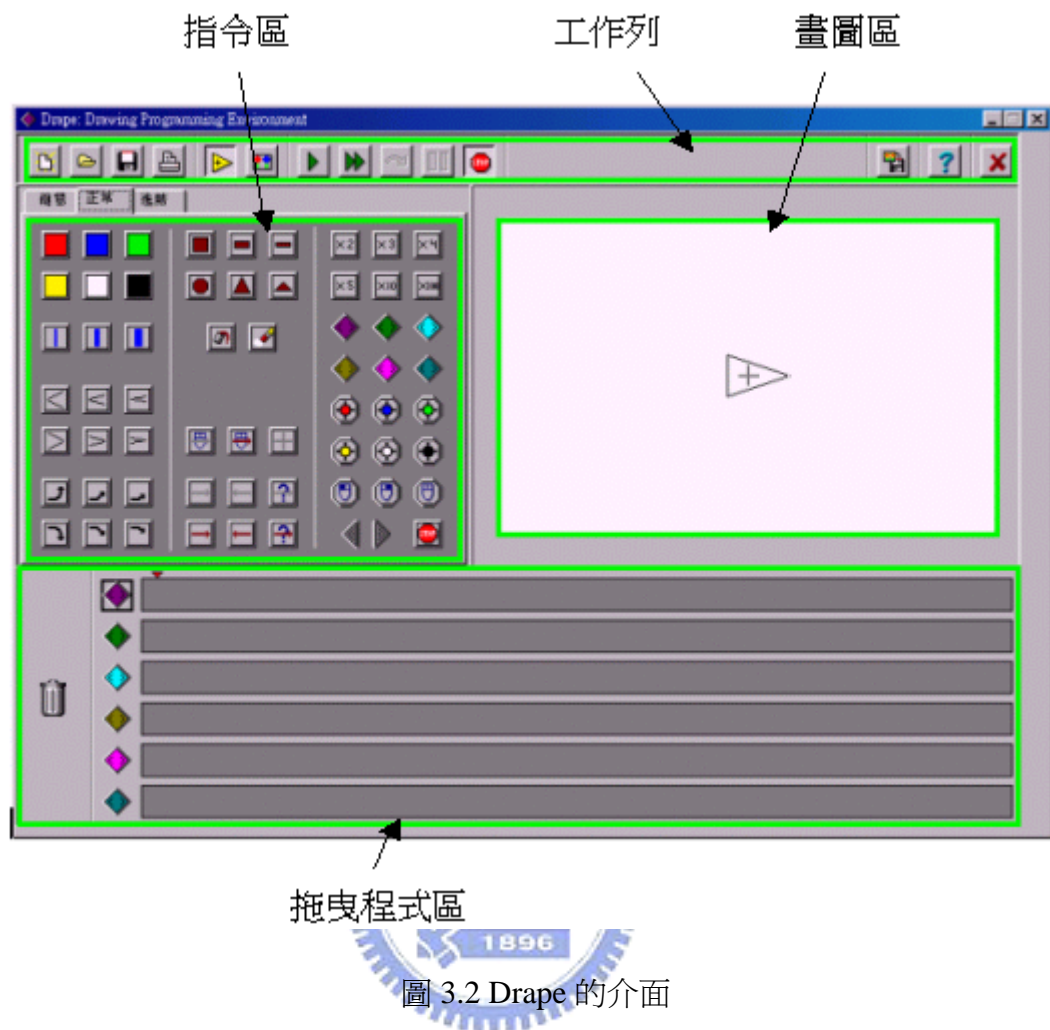


圖 3.2 Drape 的介面

白色部分是畫圖區，其他部分可以區分為

- 1.工作列：左邊有載入、儲存程式有關的按鈕，中間是與執行、停止程式有關的按鈕，右邊是存成圖檔、說明以及離開按鈕。
- 2.指令區：分為簡單、正常、進階標籤頁，有許多指令圖示，這些圖示代表不同的指令，可以用滑鼠拖曳的方式置入程式拖曳區中。
- 3.拖曳程式區：由六個灰色長條組成，可以置放被拖曳的指令集以形成程序。每個程序左邊有一個鑽形的代表圖示，其中有一個鑽形是被方形黑框包圍的，代表程式執行起點。

三、程式的執行：

程式寫好之後要執行程式時，先指明要從程式的哪個部分開始執行，預設的程式執行起點是最上方那一個鑽形，你也可以用滑鼠點選鑽形改變程式執行起點，接著按執行程式按鈕以獲得程式執行的結果，程式執行的模式一共有三種(執行(慢)、執行(快)、單

一步驟)，將在工具列上的指令作介紹。

四、增加及刪除指令。

在拖曳程式區中若指令放錯位置，只要用滑鼠將其拖曳到新位置，它就會在那裡了。若你想要移除一個指令，將它拖曳到左邊的垃圾桶，最後，當你選擇指令時，你可壓下<Ctrl>鍵來複製它。

五、工具列上的指令說明，如表 3.3。

表 3.3 工具列上的指令說明

圖示	意義	說明
	開新程式	清除現有的程式並開始新的程式。
	載入程式	載入 Drape 程式(副檔名為.drp 檔)。
	儲存程式	儲存 Drape 程式(副檔名為.drp 檔)。
	列印程式	開啓程式的文字說明視窗，可選擇列印程式或離開。
	顯示游標	按一下游標消失，再按一下又出現游標。
	全螢幕	按一下出現全螢幕，再按一下又回復有指令集的小螢幕，注意畫圖區的圖案將依螢幕大小而被縮小或放大。
	執行(慢)	緩慢地執行程式。
	執行(快)	快速地執行程式。
	單一步驟	一步步執行程式，每按一次按鈕程式執行一個步驟。
	暫停	暫時中止執行程式，再次執行程式時會從程式指令暫停點往下執行。
	停止	中止執行程式，再次執行程式時會重新開始。
	存圖至檔案	按此鈕開啓另存新檔對話框，可將當時畫圖區的圖案存成 Bitmap 圖檔。
	資訊	註明版本及作者。
	離開	離開並關閉 Drape。先要停止程式才能使用。

六、指令區的指令：

指令依其難易度而被置於簡單(Easy)、正常(Normal)與進階(Advanced)三種標籤頁中，以下一一介紹(有些指令在不同標籤頁中會重複出現，這些指令不重複介紹)。

1.簡單(Easy)標籤頁下的指令如表 3.4。


表 3.4 簡單(Easy)標籤頁下的指令

圖示	說明
	畫筆顏色的設定，基本有六種顏色，作用的範圍在下次顏色設定前。
	縮放游標前進(後退)一步的長度，<代表放大(2 倍與根號 2 倍)，>代表縮小(1/2 倍與 1/(根號 2)倍)，線寬不受影響。
	游標左轉 90、45 度以及游標右轉 90、45 度。
	在游標處畫實心正方形、長方形、扁長方形、圓形、三角形與扁三角形，顏色依顏色設定，尺寸依縮放指令而定
	填滿封閉區間以及清除畫圖區圖像。
	移動游標前進(後退)一步，方向依游標當時的指向而定。
	移動游標前進(後退)一步並畫線。
	重覆下一個指令若干次，可以累計，如 x2x3 放一起總共是重覆 6 次。
	呼叫程序，不同顏色的鑽形代表程式拖曳區中六個不同的程序，程式執行到此時會跳到指定的程序中繼續執行。
	區塊開始及結束。將多個指令集成一體，如說重覆指令只重覆下一個指令，但若以區塊將指令包起來，則其中所有的指令都將被重覆，類似數學中的括號()。

2.程式實例：


程式一：



說明	執行結果
第一個指令指定畫一條線，下一個指令將方向旋轉 90 度，再下一步畫第二條線，依此類推。	


程式二：



說明	執行結果
第一個指令設定顏色為藍色，接下來畫一個圓，第三個指令將游標位置移到右邊，畫另一個圓，依此類推。	

程式三：



說明	執行結果
在一個藍色正方形中畫一個黃色圓，再在圓中畫一個紅三角。	









程式四：






說明	執行結果
第一個、第二個指令設定重覆區塊次數(8次)，意即(旋轉 45 度，畫線)重覆 8 次，接下來的指令是旋轉 90 度，前進一步，設定黃色，填滿。	

3.正常(Normal)標籤頁下的指令如表 3.5。

表 3.5 正常(Normal) 標籤頁下的指令


圖示	說明
	筆寬，有分細、中、粗三種。
	放大 1.05 倍以及縮小 1/1.05 倍。
	
	左轉 6 度以及右轉 6 度。
	
	游標移至螢幕中心點。
	任意、無規則地移動。
	任意、無規則地畫線。

	若游標下的顏色為指定顏色，則執行下一個指令。可指定六種顏色。
	依次為若按滑鼠左鍵、若按滑鼠右鍵、若不按鍵則執行下一個指令。
	停止執行程式。

4.程式實例：


程式五：



說明	執行結果
指定用比較粗的畫筆來畫線。	

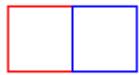
程式六：



說明	執行結果
上面的程式在一個正方形內部畫一些正方形。第一個指令是重複 10 次，每畫一次正方形，尺寸即稍微縮小一點。	


程式七：



說明	執行結果
第二行是畫正方形，而由第一行控制，第一行的意思是用紅色畫正方形，前進一步，再用藍色畫正方形。	

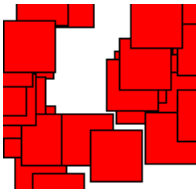
程式八：



說明	執行結果
畫筆前進一步，旋轉 45 度，縮小一點。重覆上述指令 100 次。	

程式九：



說明	執行結果(參考)
游標移動到任意的的位置，畫一個正方形。重覆上述的指令 100 次。	



程式十：



說明	執行結果(參考)
游標移動到任意的的位置，再畫到任意的位置。重覆上述的指令 100 次。	

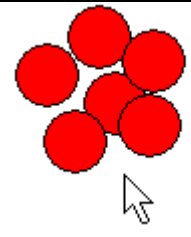
程式十一：



說明	執行結果(參考)
若按滑鼠左鍵，則畫到滑鼠游標，若不按鍵，則移動到滑鼠游標。程式將重覆上述的指令 100 萬次之後結束，程式執行時若按滑鼠左鍵拖曳即可如畫筆般畫線，請用  執行。	

程式十二：



說明	執行結果(參考)
遞迴程式(自己呼叫自己)，當你按下滑鼠左鍵，它會四處畫圓，而當你按下滑鼠右鍵，它才會停止，否則程式將持續執行。	


5.進階(Advanced)標籤頁下的指令—設定與顯示變數如表 3.6，請注意每個方形指令只有一種設定值；圓形指令可以自由修改設定值，在拖曳程式區中的圓形指令可按滑鼠右鍵修改設定值。

表 3.6 進階(Advanced)標籤頁下的指令—設定與顯示變數

圖示	說明	
	設定變數一個值 要填入變數名稱以及一個值(可以用數學式子表達)。注意變數名稱只能用文字，不能使用保留字(下面解釋)或數字，注意大小寫。	
	顯示變數 輸入變數名稱並執行後顯示變數的值，通常會連同變數名稱一起顯示，若勾選絕對，則僅顯示變數的值，輸入精度值可控制顯示變數的精度。右圖指定顯示變數 a 的名稱及值到小數點下三位。	
函數	+：加，-：減，*：乘，/：除， ：絕對值，#：四捨五入，@：cos 函數，\$：sin 函數，!：arctan 函數，^：冪次方，~：小數點以下無條件捨去。	

Drape 可以執行計算機的功能：使用設定變數，輸入變數名稱及運算式，再使用顯示變數，執行之後即可得到答案。表 3.7 介紹其計算方法。





表 3.7 變數計算方法




程式碼	程式列印	結果	說明
	將變數 a 設為 2-3*5 顯示變數 "a"的名稱及整數值	a=13	運算規則為先乘除後加減
	將變數 a 設為 3*2^0.5/2 顯示變數 "a"的名稱及值到小數點下 3 位	a=2.121	運算規則為先乘冪後乘除
	將變數 a 設為 2^(3-2.5) 顯示變數 "a"的名稱及值到小數點下 5 位	a=1.41421	運算規則為先執行括號運算

Drape 變數可設定數學函數運算，用一個符號代表函數，如#代表四捨五入計算，執行#(2.6)後得到 3，@代表 cos 函數計算，執行@(60)後得到 0.5 等等，它的運算規則完全符合數學運算順序。除此之外，變數也可以設定為已經宣告過的變數，例如將 a 設定為 3，將 b 設定為 3*a/4，執行之後得到 b 等於 2.25。

6.進階(Advanced)標籤頁下與繪圖環境有關的圖示說明，如表 3.8。

表 3.8 進階(Advanced)標籤頁下與繪圖環境有關的圖示說明

圖示	說明
	設定顏色 使用此指令你可以從更多的顏色集合中選擇顏色。
	設定線寬 你可設定線寬到任意值。
	設定角度 可設定任意角度，角度單位是度 (degree)，逆時針，比方說輸入 90 代表向左轉 90 度，通常最後的結果為游標現有角度再加上輸入的數值，例如在游標角度 90 度時輸入設定角度 30，則會得到 120 度角。你可以有另一種選擇：勾選『絕對』(absolute)的核取方塊，這樣你鍵入的值將成為真正的角度，這樣你在游標角度 90 度時輸入設定角度 30，你會得到 30 度角，而不是 120 度角。
	改變尺寸大小 此圖示可以放大或縮小圖形尺寸，輸入的值就是改變的百分率((改變後尺寸/原尺寸)*100%)，例如輸入 200 代表將尺寸放大兩倍，50 代表將尺寸縮為一半，這些改變是相對於現有尺寸，若要得到絕對尺寸，勾選『絕對』(absolute)的核取方塊即可，例如輸入 100 並勾選『絕對』(absolute) 即可得到電腦預

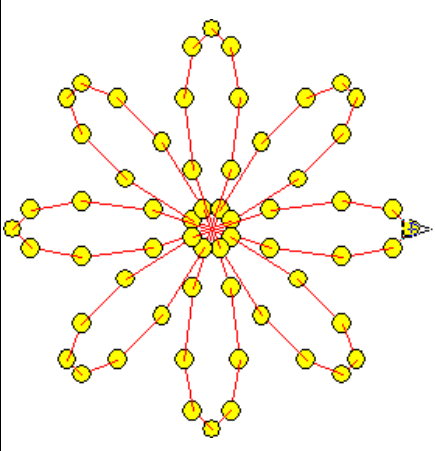
	設的尺寸設定。
	<p>移動到一個新位置</p> <p>輸入 a, b 參數使游標移動到平面座標系的點(a, b)上，同樣有兩種選擇：勾選『絕對』(absolute)的核取方塊，游標會以絕對座標位置來移動，所謂絕對座標是以畫圖區的中心座標為$(0, 0)$，x 軸為水平線，y 軸為垂直線而訂出來的座標系統，此系統與游標現有的角度及位置無關；若不勾選『絕對』(absolute)的核取方塊，則游標會以相對座標位置來移動，所謂相對座標是以現有游標位置為原點，x 軸為過原點而平行游標指向線之直線，y 軸為過原點而垂直游標指向線之直線而訂出來的座標系統。注意移動的單位長度是依目前的尺寸設定來決定。</p>
	<p>畫線到一個新位置</p> <p>同上，但這次畫線到一個新位置上。</p>
	以指定的次數重複下一個指令

7.程式實例：

程式十三：以極座標的觀點來畫花瓣線。




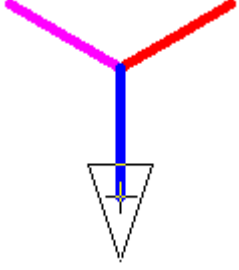
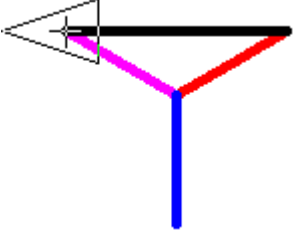
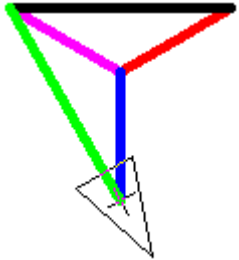
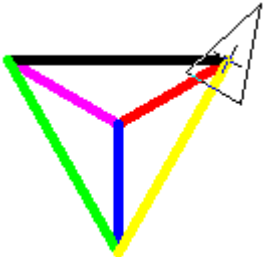
列印程式	說明	結果
<p>程序 1</p> <p>程序開始</p> <p>設定相對尺寸比率為 20%</p> <p>將變數 n 設為 4</p> <p>將變數 r 設為 2</p> <p>將變數 th 設為 0</p> <p>移動到絕對座標(X, Y)： $(r, 0)$</p>	<p>1. 將變數設為 4，如下圖：</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p>將變數： <input type="text" value="n"/></p> <p>設定成： <input type="text" value="4"/></p> </div> <p>2. 移動到絕對座標，須在絕對核取框中打勾。</p>	<p>設定相對尺寸為 20%，表示畫的實心圓會變為原來的 $1/5$，n 與花瓣的個數有關，r 是花瓣的半徑，th 是角度。可發現游標縮小，並移動到第一個位置$(r, 0)$，因為程式以絕對座標繪圖，故尺寸縮小不會影響畫線的軌跡。</p>

<p>重覆 61 次</p> <p>區塊開始</p> <p>將變數 fr 設為 $r * \sin(n * th + 180 / 2)$</p> <p>畫線到絕對座標(X,Y) : $(fr * \cos(th), -fr * \sin(th))$</p> <p>顏色設定黃色</p> <p>畫圓</p> <p>顏色設定紅色</p> <p>將變數 th 設為 $th + 6$</p> <p>區塊結束</p> <p>程序結束</p>	<p>重覆區塊 61 次，區塊中設定：</p> <p>$fr = r * \sin(n * th + 180 / 2)$</p> <p>畫紅線到點座標：$(fr * \cos(th), -fr * \sin(th))$，並在此點畫一黃色圓(回簡單或正常標籤頁下選取)。</p> <p>每重覆一次區塊，角度就增加 6 度，重覆 60 次就繞了一圈，但為了使圖形首尾相接，重覆區塊 61 次。</p>	
--	--	--

程式十四：以相對及絕對座標的觀點來畫出一 Y 字形並連接端點。



列印程式	說明	結果
<p>程序 1</p> <p>程序開始</p> <p>設定線寬為 5</p> <p>設定游標相對角度 30 度</p> <p>向前畫線一步</p> <p>向前畫線一步</p> <p>移到中心</p>	<p>設定畫線寬度為 5。</p> <p>游標左轉 30 度(不勾選絕對核取方塊)，向前畫線兩步之後移回中心點(此時游標角度仍為 30 度)。</p>	
<p>改變顏色為 16711935</p> <p>設定游標相對角度 120 度</p> <p>畫線到相對游標座標(X,Y) : (2, 0)</p>	<p>點選顏色(挑基本色彩最右列第二行色彩)：</p> 	<p>游標再轉 120 度，與 x 軸夾 150 度角。</p> 

顏色設定藍色 移動到絕對座標(X,Y)：(0,0) 設定游標相對角度 120 度 畫線到相對游標座標(X,Y)：(2,0)	游標移動到絕對座標 0,0 點代表至畫圖區的正中央，游標再轉 120 度，與 x 軸夾 270 度角，向游標指向前進兩步並畫線。	
顏色設定黑色 移動到絕對座標(X,Y)：(3 ^{0.5} ,1) 設定游標相對角度 270 度 畫線到相對游標座標(X,Y)：(2*3 ^{0.5} ,0)	游標移到絕對座標(3 ^{0.5} ,1)，這是第一條線(紅線)的端點。 游標再轉 270 度，與 x 軸夾 180 度角，畫線到第二條線(粉紅線)的端點。	
顏色設定綠色 設定游標相對角度 120 度 畫線到相對游標座標(X,Y)：(2*3 ^{0.5} ,0)	游標再轉 120 度，與 x 軸夾 -60 度角，畫線到第三條線(藍線)的端點。	
顏色設定黃色 設定游標相對角度 120 度 畫線到相對游標座標(X,Y)：(2*3 ^{0.5} ,0) 程序結束	游標再轉 120 度，與 x 軸夾 60 度角，畫線到第一條線(紅線)的端點。	

8.Drape 的保留字：表 3.9 列出 Drape 的保留字，為寫程式提供另一種選擇。

表 3.9 Drape 的保留字

保留字	意義	保留字	意義
x	游標 x 座標	y	游標 y 座標
angle	游標角度	size	尺寸大小
linewidth	筆寬	color	顏色
pwidth	畫圖區寬度	pheight	畫圖區高度
xmouse	滑鼠游標的 x 座標	ymouse	滑鼠游標的 y 座標
ccolor	游標下的顏色	large	大螢幕(1)或小螢幕(0)播放

lmouse	是否滑鼠左鍵被壓下	rmouse	是否滑鼠右鍵被壓下
random	亂數	pi	圓周率
frozen	程式執行完畢後才呈現圖形，加快畫圖速度		

9.程式實例：

程式十五：顯示 Drape 內定的保留字值。



列印程式	說明
(1)移動到相對游標座標(X,Y)： (-3 , 3) 顯示變數 “x”的名稱及整數值	<p>1.列印程式中的編號(1)到(10)，是為方便操作而設，不存在於原始程式中。</p> <p>2. 顯示變數 “x”的名稱及整數值，即是顯示x的整數部分(精度須設為0)。</p> <p>3.程式以不斷重覆移動及顯示變數將部份保留字顯示出來。結果如下圖，因滑鼠游標停駐在畫圖區中心附近，故xmouse=0, ymouse= 0。</p>
(2)移動到相對游標座標(X,Y)： (3 , 0) 顯示變數 “y”的名稱及整數值	
(3)移動到相對游標座標(X,Y)： (-1 , -1) 顯示變數 “angle”的名稱及整數值	
(4)移動到相對游標座標(X,Y)： (0 , -1) 顯示變數 “size”的名稱及整數值	
(5)移動到相對游標座標(X,Y)： (0 , -1) 顯示變數 “linewidth”的名稱及整數值	
(6)移動到相對游標座標(X,Y)： (0 , -1) 顯示變數 “color”的名稱及整數值	
(7)移動到相對游標座標(X,Y)： (-2.5 , -1) 顯示變數 “pwidth”的名稱及整數值	
(8)移動到相對游標座標(X,Y)： (6 , 0) 顯示變數 “pheight”的名稱及整數值	
(9)移動到相對游標座標(X,Y)： (-6 , -1) 顯示變數 “xmouse”的名稱及整數值	
(10)移動到相對游標座標(X,Y)： (6 , 0) 顯示變數 “ymouse”的名稱及整數值	

結果：

```

x=-3  y=3
  angle=0
    size=100
  linewidth=1
    color=255
pwidth=12  pheight=8
xmouse=0  ymouse=0

```


程式十六：可執行按滑鼠左鍵點選顏色，按滑鼠右鍵畫正方形。



列印程式	說明	結果(參考)
程序 1 程序開始 顏色設定藍色 畫圓 向前移動一步 顏色設定綠色 畫圓 呼叫程序 2 程序結束	在此程序畫一藍色及綠色圓。	
程序 2 程序開始 若按下左鍵則 將變數 color 設為 ccolor 若不按鍵則 移到滑鼠游標 若按下右鍵則 畫正方形 呼叫程序 2 程序結束	在此程序按左鍵則將 Drape 游標下的顏色設定為目前顏色，不按鍵則 Drape 游標跟著滑鼠游標，按右鍵則以目前顏色畫正方形。按滑鼠左鍵在圓形圖中點選顏色，然後按滑鼠右鍵在滑鼠游標處畫此顏色之方形，注意以 執行。	

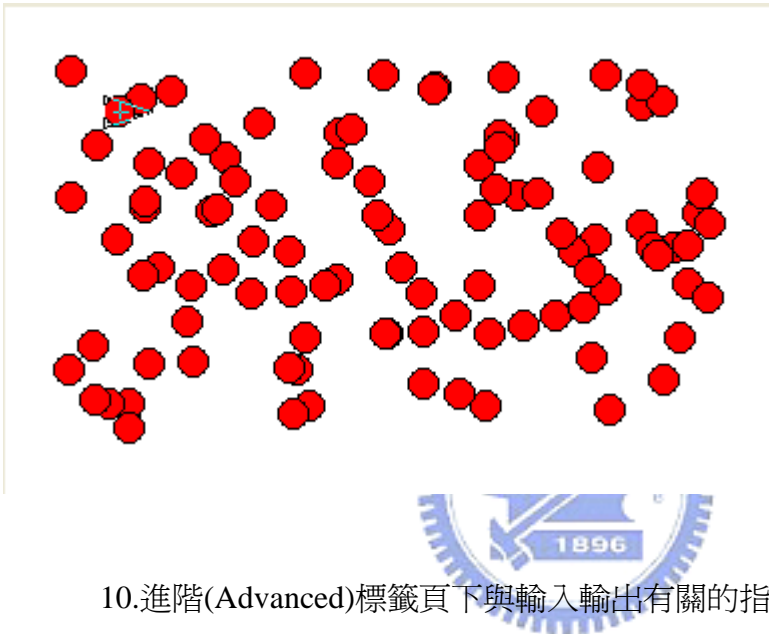
程式十七：在畫圖區任意位置畫 100 個小圓，以畫圖區邊界上下左右向內移一單位長當作畫圖範圍。



列印程式	說明
程序 1 程序開始 尺寸縮小 重複 100 次 區塊開始	在重複區間內設定的變數 x 及 y，即是畫圓步驟中的圓心座標。以 x 的設定為例， $\text{random} * \text{pwidth}$ 的值代表從 0 到畫圖區寬的亂數分布，因畫圖區的 x 座標是 $-\text{pwidth}/2$ 到 $\text{pwidth}/2$ ，使用 $\text{random} * \text{pwidth} - \text{pwidth}/2$ 代表

將變數 x 設為 random*(pwidth-2)-pwidth/2+1 將變數 y 設為 random*(pheight-2)-pheight/2+1 畫圓 區塊結束 程序結束	整個畫圖區寬度的亂數分布，爲了讓畫圖範圍左右各縮一單位，將 x 設定爲 random*(pwidth-2)- pwidth/2+1，此情形同樣適用於 y 的設定上。
--	---






結果(參考)：



10.進階(Advanced)標籤頁下與輸入輸出有關的指令，如表 3.10。

表 3.10 進階(Advanced)標籤頁下與與輸入輸出有關的圖示說明


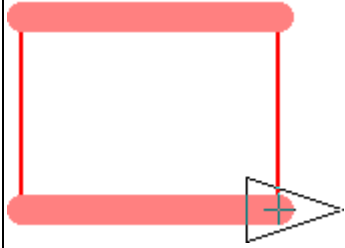
圖示	說明
	在目前位置上畫出旋轉的字體。 同上，但這一次字體角度依照目前角度旋轉。
	在目前的位置上畫一個影像 系統會問你影像檔案名稱，它可以是.bmp，.ico，.wmf 或.emf 影像。鍵入你所知道到的檔案名稱或者按下 <input type="text"/> 按鈕來尋找檔案。通常，這影像會依照目前的尺寸值而被縮放，若勾選”絕對”(absolute)的核取方塊，則影像不被縮放。影像檔案必須被存在與程式相同的檔案夾中，否則執行程式時會找不到檔案。
	將影像存入檔案 必須先設定檔案名稱以便儲存影像(它將被存成.bmp 檔)。
	播放一段音效

	系統會向你要一個音效的檔案名稱，你可以鍵入你所知道到的檔案名稱或者按下標示“...”的按鈕來尋找檔案。可以播放 wave 檔以及 midi 檔。這聲音檔案必須被存在與程式相同的檔案夾中，否則執行程式時會找不到檔案。
	休息一段時間 以毫秒(milliseconds)為單位設定休息時間。
	若是游標下的顏色符合，執行下一個步驟。 你指定一個顏色，當目前游標下的顏色與指定顏色相同時，執行下一個步驟。
	若使用者按下特定鍵，則執行下一個步驟。 在『設定一個值』對話框中輸入一個字母，執行程序經過此指令時若此字母按鍵被壓下，則執行下一個指令，否則跳過下一個指令而繼續執行程式。
	儲存現有環境及回復上一個環境。 環境變數儲存顏色、尺寸大小、線寬、角度、位置等資訊，你可將環境變數命名，之後要回復原來的環境變數，只要用回復指令並輸入原環境變數名稱即可。
	呼叫一個程式 系統會要你提供所要呼叫程式的檔案名稱，鍵入你所知道到的檔案名稱或者按下標示“...”的按鈕來尋找檔案。這程式檔案必須被存在與程式相同的檔案夾中，否則執行程式時會找不到檔案。程式呼叫可以是巢狀的，那就是說，一個程式可以呼叫另一個，這一個再依順序呼叫另外一個，諸如此類。但是有一個巢狀深度的限制，如果超過了這限制，系統會警告你。

11. 程式實例：

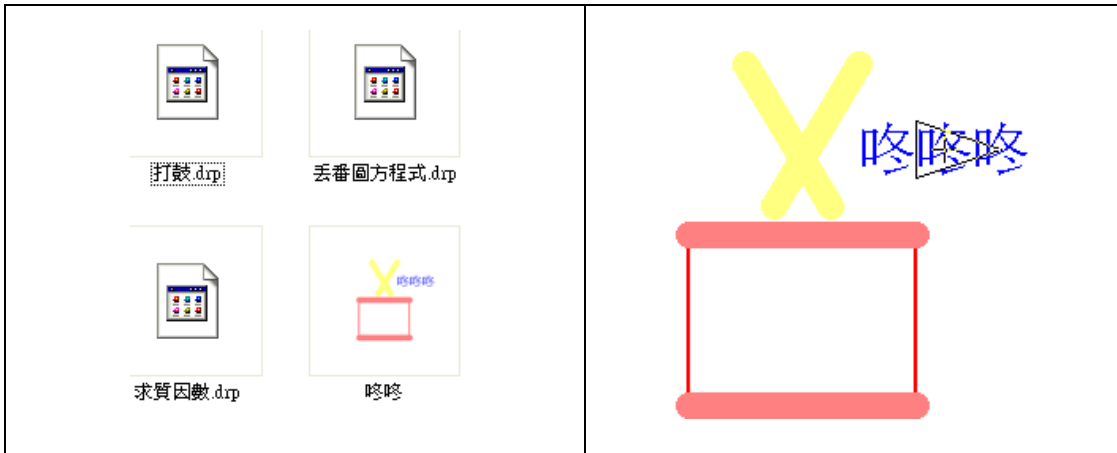
程式十八：



程式列印	說明	結果
程序 1 程序開始 設定線寬為 2 移動到相對游標座標(X,Y)：(-2,0) 畫線到相對游標座標(X,Y)：(0,-3) 移動到相對游標座標(X,Y)：(4,3) 畫線到相對游標座標(X,Y)：(0,-3)	1. 點選顏色(挑基本色彩最左列第一行色彩)： 	

<p>改變顏色為 8421631 移動到相對游標座標(X,Y)：(-4,3) 設定線寬為 15 畫線到相對游標座標(X,Y)：(4,0) 移動到相對游標座標(X,Y)：(-4,-3) 畫線到相對游標座標(X,Y)：(4,0)</p>	<p>2.先畫兩細線，再改變顏色，用粗線寬來畫水平線。</p>	
<p>改變顏色為 8454143 移動到相對游標座標(X,Y)：(-3,6) 畫線到相對游標座標(X,Y)：(1.5,-2.5) 移動到相對游標座標(X,Y)：(0.5,2.5) 畫線到相對游標座標(X,Y)：(-1.5,-2.5)</p>	<p>改變顏色後畫一個叉叉。</p>	
<p>顏色設定藍色 移動到相對游標座標(X,Y)：(3,1) 畫旋轉文字 "咚咚咚"</p>	<p>照著游標的指向列印文字。在設定一個值對話框中輸入咚咚咚。</p>	<p>文字： 咚咚咚</p> <p>咚咚咚</p>
<p>圖形存入檔案 "咚咚.bmp" 程序結束</p>	<p>使用圖形存入檔案，在輸入一個值對話框中圖檔名稱輸入：咚咚.bmp。</p>	<p>圖檔名稱： 咚咚.bmp</p>

上述的程式首先畫一個鼓以及鼓棒，在圖像旁邊印出咚咚咚三個字，最後存檔到咚咚.bmp，當執行程式後，會在此程式(打鼓.drp)相同資料夾發現圖檔咚咚.bmp 已經被產生了，瀏覽圖檔可看到程式執行的結果。



進一步的說明，由於此程式移動及畫線完全是相對座標所構成，因此若加入平移、旋轉、縮放尺寸，其相對關係依然維持不變(在拖曳程式區中須讓滑鼠游標出現箭頭，才能插入新指令)：

程式十八之一：



程式十八之二：



程式十八之三：



<p>程式十八之一 中心被移到(3,0)而開始畫圖，所以圖畫的一部分被邊界遮蓋了，只看到咚咚兩個字。</p>	<p>程式十八之二 在圖畫區中心畫縮小圖形，筆寬沒縮小，因此叉又與下面圖形會有重疊。</p>	<p>程式十八之三 畫縮小旋轉圖，文字也跟著旋轉。</p>

程式碼十九：鍵盤輸入與聲音的處理。



程式列印	說明	結果
程序 1 程序開始 若按下 "1" 鍵則 聲音播放 "DRUM1.wav" 若按下 "2" 鍵則 聲音播放 "DRUM2.wav" 呼叫程序 1	若按下數字鍵 1 則播放 "DRUM1.wav"，若按下數 字鍵 1 則播放 "DRUM2.wav"，接著再跳 回程序 1，這是一個遞迴 程式，不斷地執行檢查。	程式執行時如說明，注意 DRUM1.wav 與 DRUM2.wav 需放在與主 程式相同的檔案夾中。


稍微修改程式碼如下，得程式十九之一：



程式列印	說明	結果
程序 1 程序開始 從檔案載入圖形 "咚咚.bmp" 呼叫程序 2 程序結束	本程序會在畫圖區中心 載入"咚咚.bmp"。由於圖 檔已經縮小並旋轉，其圖 形如結果所示。	
程序 2 同程式十九 略。	說明同程式十九。	結果同程式十九。

下面的程式可透過滑鼠點擊指定顏色來播放聲音。程式十九之二：



程式列印	說明	結果
程序 1 程序開始 設定相對尺寸比率為 70% 從檔案載入圖形 "咚咚.bmp" 呼叫程序 2 程序結束	將"咚咚.bmp"圖檔尺寸縮 小為原尺寸的 0.7 倍，其 他說明同程式十六之一。	

程序開始 若按下左鍵則 區塊開始 移到滑鼠游標 若顏色為 8421631 則 聲音播放 "DRUM1.wav" 若顏色為 8454143 則 聲音播放 "DRUM2.wav" 休眠 200 毫秒 區塊結束 呼叫程序 2	在按下滑鼠左鍵時使用 移到滑鼠游標指令並判 定游標下的顏色。若為指 定顏色則播放鼓聲。此程 序表示在滑鼠點選時若 游標下的顏色為 8421631 (鼓面顏色)則播放鼓聲， 若游標下的顏色為 8454143 (鼓棒顏色)則播 放另一鼓聲。	略。
---	---	----

最後，環境變數存放了目前的位置、角度以及尺寸大小。設計程式時常常需要回復到前一個環境設定，與其將位置、角度以及尺寸大小等重新設定回來，不如在呼叫這部分之前先儲存環境，並在之後回復此環境。

12.進階(Advanced)標籤頁下與變數控制有關的指令，如表 3.11。

表 3.11 與變數控制有關的圖示說明：

圖示	說明
	變數比較結果是否為真 使用此圖示時電腦會請你輸入一個情況，根據情況的真假來執行程序，用數學上的運算元： $<$, $>$, $=$, $<=$, $>=$, $<>$ (不等於)來操作。例如，輸入 $x > xmouse$ 或者 $color = red$ ，若結果為真，則執行下一步驟(區塊)，若結果為假，則跳過下一步驟(區塊)。有時會有多重情況要判別，請將不同情況用括號括起來，然後使用加號，代表將兩個情況以或(or)連接，乘號代表將兩個情況以且(and)連接。例如 $(a=1)+(a=2)+(a=3)$ 代表若變數 a 為 1 或 2 或 3，例如 $(x > -pwidth/2) * (x < pwidth/2)$ 則代表若 x 的位置在畫圖區中則執行下一個指令。
	離開目前的程序：此指令結束現有程序，跳回上一程序繼續執行。

13.程式實例：

程式二十：示範倒數計時，由數字 10 開始，0.5 秒為間格而往下計數。



流程圖	列印程式	說明
<pre> graph TD Start([開始]) --> Init[變數設定 n=0 變數設定 i=10] Init --> Decision{i < n} Decision -- 否 --> Loop[顯示變數 i 變數設定 i=i-1 休眠 500 毫秒 清除影像] Loop --> Decision Decision -- 是 --> EndPrint[列印『程式結束』] EndPrint --> End([結束]) </pre>	<p>程序 1 程序開始 將變數 n 設為 0 將變數 i 設為 10 呼叫程序 2 畫旋轉文字 "程式結束" 程序結束</p> <p>程序 2 程序開始 若變數為 $i < n$ 則 離開目前程序 顯示變數 "i" 的值取整數 將變數 i 設為 i-1 休眠 500 毫秒 清除影像 呼叫程序 2 程序結束</p>	<p>第一行程序為初始值設定以及輸出，第二行作變數處理以及清除影像動作，使數目字不會重疊而漸次呈現。使用『離開目前程序指令』，跳回程序 1，印出程式結束後終止。</p>

程式二十一：示範數字的連加，計算 $1+2+3+4+\dots+10$ 的結果。



流程圖	列印程式	說明
<pre> graph TD Start([開始]) --> Input[/輸入 n=10/] Input --> Init[i=1; sum=0] Init --> LoopBody[sum=sum+i; i=i+1] LoopBody --> Decision{i <= n} Decision -- 是 --> LoopBody Decision -- 否 --> Display[/顯示 sum/] Display --> End([結束]) </pre>	<p>程序 1 程序開始 將變數 n 設為 10 將變數 i 設為 1 將變數 sum 設為 0 呼叫程序 2 程序結束</p> <p>程序 2 程序開始 將變數 sum 設為 sum+i 將變數 i 設為 i+1 若變數為 $i \leq n$ 則 呼叫程序 2 顯示變數 "sum" 的名稱 及整數值 程序結束</p>	<p>將連加法通式擴充為 $1+2+3+\dots+n$，若要計算 1 加到 10 的總和，設定 $n=10$ 即可。i 變數每次加 1，sum 變數設定為 sum+i。一直算到 $i=n$ 為止。</p>

第四章 開發解決問題之應用程式範例

本章以六個程式呈現 Drape 程式可以應用的範圍：韓信點兵與求餘數有關；聯立方程式運用縮小解答範圍以求得精確解答；產品設計運用 Drape 變數計算與繪圖，呈現罐頭在表面積固定下，容量的極值問題；牛頓拉夫森法是以 Drape 解決數值問題的探討；蒙地卡羅法是以 Drape 中的亂數與顏色判定，探討面積大小；方格法以另一個角度，類似掃描的方式來求出面積大小。

第一節 韓信點兵(程式碼見附錄一)：

一、概述：

『物不知數問題』也稱孫子問題（載孫子算經卷下二十六：『今有物不知其數，三三數之剩二，五五數之剩三，七七數之剩二，問物幾何。』『答約二十三』。）這個問題需求解一次同餘式，後來孫子問題成爲一種廣泛流傳的民間數學遊戲，被稱爲『韓信點兵』。此問題屬於不定方程的應用，用一般代數方程組表示如下：

$$x = 3q_1 + 2$$

$$x = 5q_2 + 3$$

$$x = 7q_3 + 2$$

x 是所求的數， q_1 、 q_2 、 q_3 表示 x 分別被 3、5、7 除所得的商數，其中 x、 q_1 、 q_2 、 q_3 爲正整數或 0。答案有無窮多組，要求正整數解。

二、程式設計流程圖：

這個問題是要找出某數(被除數)、除數與餘數之間的關係，由於
被除數=商數*除數+餘數，知
餘數=被除數-商數*除數

上式的商數是被除數/除數後取最大整數之結果。

以某數「三三數之剩二，五五數之剩三，七七數之剩二」而言，某數須同時滿足「除以 3 餘 2，除以 5 餘 3，除以 7 餘 2」三個條件。在 Drape 程式中可設定 $a=3$ 、 $b=2$ 、 $c=5$ 、 $d=3$ 、 $e=7$ 、 $f=2$ ，其中 a、c、e 爲除數，b、d、f 爲餘數。變數 start 除以 a 的餘數，由上面餘數的定義，可以寫成 $\text{start} \sim (\text{start}/a)*a$ 。若變數 start 同時符合 $b=\text{start} \sim (\text{start}/a)*a$ ， $d=\text{start} \sim (\text{start}/c)*c$ 且 $f=\text{start} \sim (\text{start}/e)*e$ ，此時 start 就爲其中一解。本程式要求在 1 到 6000 找尋符合條件的解。此程式流程圖如圖 4.1。

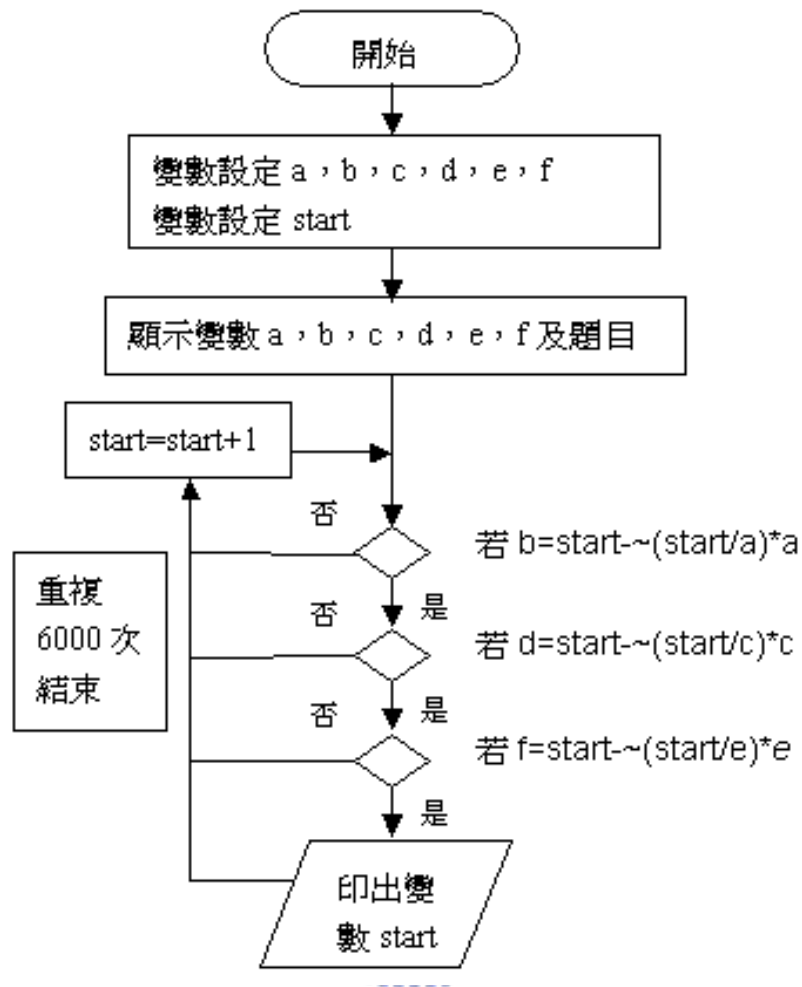


圖4.1 韓信點兵程式設計流程圖

三、程式的執行：

以某數除以 3 餘 2，除以 5 餘 3，除以 7 餘 2 求某數解，執行情形如圖 4.2

韓信點兵，每 $a=3$ 個一數，剩下 $b=2$ 個，
每 $c=5$ 個一數，剩下 $d=3$ 個，每 $e=7$ 個一數，
剩下 $f=2$ 個，問原有兵丁數可能為幾個？

兵丁數算範圍： $start \rightarrow start+6000$ 。你可設定 $start$ 變數當作數算起始點。

答案： 863
23 ~~963~~
128
233
338
443
548
653
758

圖 4.2 韓信點兵程式執行情形

程式會顯示 1 到 6000 內符合條件的解。

又設若某數除以 4 餘 1，除以 5 餘 3，除以 7 餘 2 求某數解。重新設定 a、b、c、d、e、f 值，再執行結果，如圖 4.3。

韓信點兵，每 a=4 個一數，剩下 b=1 個，
每 c=5 個一數，剩下 d=3 個，每 e=7 個一數，
剩下 f=2 個，問原有兵丁數可能為幾個？

兵丁數算範圍：start→start+6000。你可設定start變數當作數算起始點。

答案：	1213	2473	3733	4993
	93	1353	2613	3873
	233	1493	2753	4013
	373	1633	2893	4153
	513	1773	3033	4293
	653	1913	3173	4433
	793	2053	3313	4573
	933	2193	3453	4713
	1073	2333	3593	4853

圖 4.3 重設參數後再執行韓信點兵程式結果

本程式可以求三個條件的韓信點兵問題，除了可以設定 a、b、c、d、e、f 值之外，還可以設定變數 start 值，來改變數算的起點。由參數改變以及模擬，可看出此種不定方程式解的特徵，即所有解形成一等差級數，圖 4.3 顯示，93、233、373、513... 是一等差級數，公差為 140，恰為除數 4、5、7 的最小公倍數，藉由程式，可以很快察覺這些解的特性。

第二節 聯立方程式(程式碼見附錄二)：

一、概述：

本範例在求兩個二元一次方程式 $y=a*x+b$ 與 $y=c*x+d$ 的解，考慮兩方程式的圖形，滿足 $y=a*x+b$ 的所有 x 、 y 值形成了直角座標上的一條直線， $y=c*x+d$ 形成了另一條直線，若此兩條直線有交點，則此交點座標的 x 、 y 值既滿足前式，又滿足後式，為此兩式的通解。

要解出交點座標 x 、 y 值，可以先取 x 軸的一段，分割此段為若干區間，並判別區間內有無交點，考慮通過點 (x_a, e) 、點 (x_b, g) 的一條直線，與通過點 (x_a, f) 、點 (x_b, h) 的另一條直線如圖 4.4。

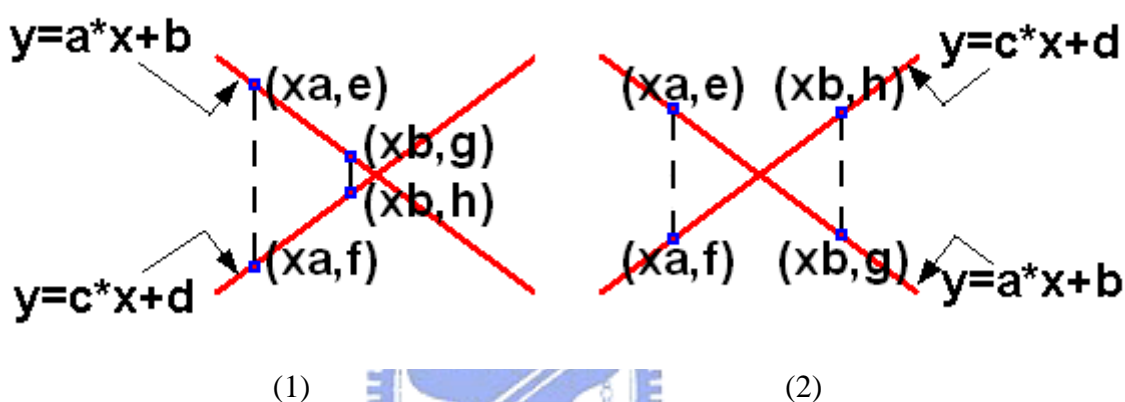


圖 4.4 在 $[x_a, x_b]$ 中無交點(1)與在 $[x_a, x_b]$ 中有交點(2)的示意圖

我們不難發現 g 與 h 的大小會在經過交點時反轉過來，即原來大的值變小，小的值會變大，所以下列判別式可以判定區間內有無交點：

1. 若區間內無交點，則 $(e-f)*(g-h) > 0$ 。其中 $e=a*x_a+b$ ， $f=c*x_a+d$ ， $g=a*x_b+b$ ， $h=c*x_b+d$ 。
2. 若區間內有交點，則 $(e-f)*(g-h) < 0$ 。
3. 若交點恰在區間端點上， $(e-f)*(g-h) = 0$ 。

找出交點判別式後，就可以找出交點所在區間，再分割此區間，找交點所在區間，如此反覆進行，就可以得到相當精準的解。

二、程式設計流程圖：

設定變數 a, b, c, d ，就是 $y=a*x+b$ 以及 $y=c*x+d$ 的係數。找尋交點的區間可以調整，但為求簡化，初始區間範圍暫定 $x_a=-100$ ， $end=100$ ，每次的增量為 $inc=(end-x_a)/1000$ 。這裡增量不直接設成常數，因為每次程式執行後會把交點存在的範圍縮小，重新設定範圍 x_a 與 end ，會得到更小的增量，更有機會找到交點。

求第一個區間的另一端點： $x_b=x_a+inc$ ，接著計算 $[x_a, x_b]$ 區間中的 y 值：
 $e=a*x_a+b$ ， $f=c*x_a+d$ ， $g=a*x_b+b$ ， $h=c*x_b+d$ 。接著判斷若 x_b 點為交點 x 值 $((e-f)*(g-h)=0)$ ，

則印出 xb 與 $a*xb+b$ 的值並結束程式。

若 $(e-f)*(g-h)<0$ ，代表 $[xa,xb]$ 間有交點，印出 xa 與 xb 。否則設定 $xa=xb$ (區間向右移動)，再計算 e,f,g,h 值來找交點。隨著區間範圍縮小，可以迅速地找到交點，本範例設定求解的精確度到小數點下第 8 位。

程式流程圖如下：

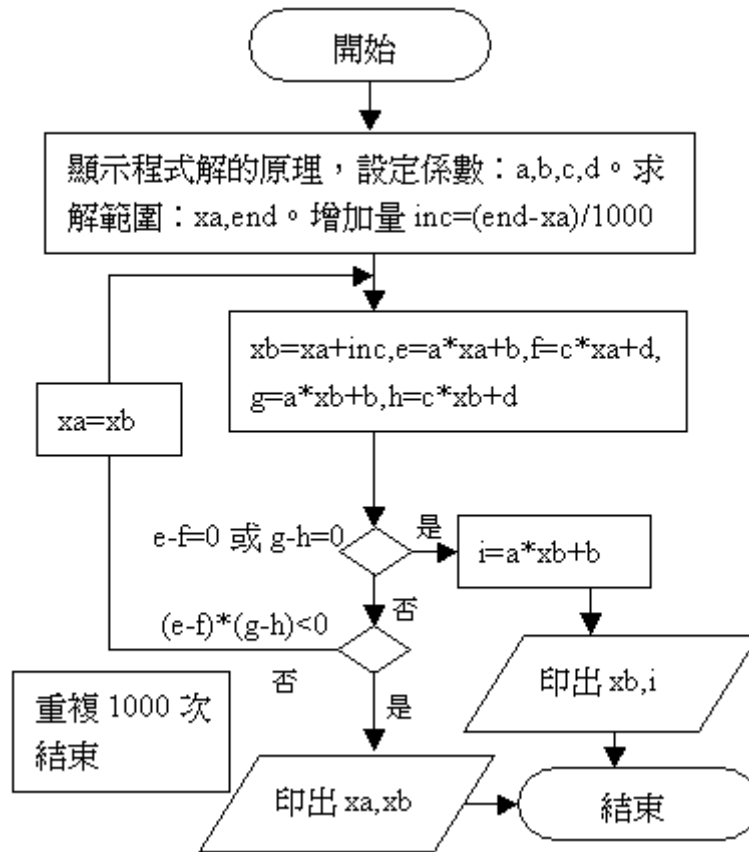


圖4.5 聯立方程式設計流程圖

三、程式的執行：

首先呈現聯立方程式原理說明，以兩頁播放，看完第一頁可以按下一頁跳到第二頁，第一頁如圖 4.6，介紹聯立方程式與交點的意義，第二頁如圖 4.7，以圖說明在什麼情況下區間內會有交點，可以用此特徵找出交點所在的範圍。

解聯立方程式 下一頁

在座標平面上的 $y=a*x+b$ 代表直線方程式， $y=a*x+b$ 與 $y=c*x+d$ 並列代表不同的兩直線方程式，也稱為聯立方程組。解聯立方程的意思就是要找兩直線的交點。本例由第四行啓動，尋找特定區間內是否有交點。

上一頁 考慮兩直線在區間 $[x_a, x_b]$ 內的情形，以 e, f, g, h 代表Y值如圖：

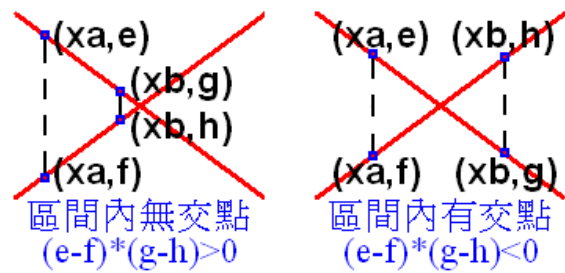


圖 4.6 聯立方程式說明(1)

圖 4.7 聯立方程式說明(2)

輸入 $a=3, b=6, c=-4, d=-2$ ，執行過後，得到交點的範圍在-1.2 與-1 之間。將左邊範圍-1.2 代入 x_a ，-1 代入 end 再執行後得到交點的範圍在-1.143 與-1.1428 之間，如圖 4.8。

解聯立方程式：

$$y = (3) * x + (6)$$

$$y = (-4) * x + (-2)$$

解 x 的範圍：

$$-1.2 \quad -1$$



解聯立方程式：

$$y = (3) * x + (6)$$

$$y = (-4) * x + (-2)$$

解 x 的範圍：

$$-1.143 \quad -1.1428$$

圖 4.8 聯立方程第一次、第二次求解

將左邊範圍-1.143 代入 x_a ，-1.1428 代入 end 再執行後得到交點的範圍在-1.1428572 與-1.142857 之間，將左邊範圍-1.1428572 代入 x_a ，-1.142857 代入 end 再執行後得到解為-1.14285714 與 2.57142857，如圖 4.9。

解聯立方程式：

$$y = (3) * x + (6)$$

$$y = (-4) * x + (-2)$$

解 x 的範圍：

$$-1.1428572 \quad -1.142857$$

解聯立方程式：

$$y = (3) * x + (6)$$

$$y = (-4) * x + (-2)$$

解為： -1.14285714 2.57142857

圖 4.9 聯立方程第三次、第四次求解

本範例利用交點判別式來判定區間內是否有交點，將一個固定範圍分割成若干區間，若區間內有交點，且交點恰在區間端點，則直接印出交點，否則再分割區間，一直到區間端點與交點的距離小於 10^{-8} 為止。另一方面，若在選取的範圍內找不到交點，則程式也會顯示沒有交點的訊息。聯立方程式是中學生必學的數學概念，本程式可以提供解聯立方程式的另類思維。



第三節 產品設計(程式碼見附錄三)：

一、概述：

在產品設計時需考慮成本問題。以鳳梨罐頭為例，通常設計師會希望能以最少的罐頭材料，包裝最多的鳳梨，以節省包裝成本。本範例探討如何設計最低成本的圓柱形罐頭，也就是在固定罐頭材料重量之下，什麼條件可以形成最大的包裝體積？因為罐頭材料重量等於密度*厚度*表面積，假定罐頭材料的密度是定值，各部分的厚度也相同，則上述問題可以化簡在固定罐頭表面積之下，什麼條件可以形成最大的包裝體積？這與數學上的極值問題相關。

首先推得一般通式：

因為算數平均數大於等於幾何平均數

$$\text{所以 } a^3+b^3+c^3 > 3a*b*c, (a^3+b^3+c^3)/3 \geq (a^3*b^3*c^3)^{1/3} \text{ -----通式}$$

極值發生於 $a^3=b^3=c^3$ 時，此時左式等於右式。

考慮一個表面積為 A，體積為 V 的罐頭如圖 4.10：

表面積的算法為： $A=2*PI*r^2+2*PI*r*h$ ，

設若 $a^3=2*PI*r^2$ ， $b^3=c^3=PI*r*h$

代入上面通式：

$$(a^3+b^3+c^3)/3 = A/3 \geq (a^3*b^3*c^3)^{1/3}$$

$$A/3 \geq [(2*PI*r^2)*(PI*r*h)*(PI*r*h)]^{1/3}$$

$$= (2*PI^3*r^4*h^2)^{1/3}$$

$$= [(PI*r^2*h)^2*(2*PI)]^{1/3}$$

$$= [V^2*(2*PI)]^{1/3}$$

故表面積固定，當 $a^3=b^3=c^3$ 時體積有極大值，即：

$$2*PI*r^2 = PI*r*h, h=2*r \text{ 時，亦即 } h/r=2 \text{ 時體積有極值。}$$

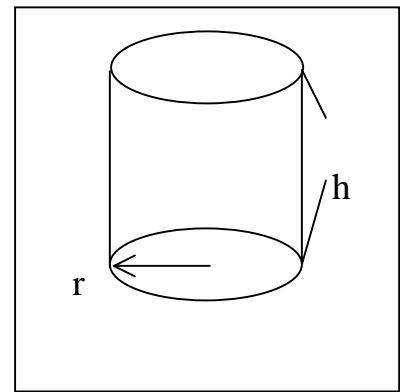


圖 4.10 罐頭示意圖

二、程式設計流程圖：

另一方面，程式模擬罐頭尺寸，將面積、半徑設成變數，另外每次讓半徑增加一個增量來比較前後的體積變化，所以罐頭的底面積會隨著模擬的進行而越變越大，高度則越變越小，這裡設定若罐頭高度小於 0.2，則模擬停止。

首先我們設定面積 a，增量 i，半徑 r。

在螢幕上呈現資料表，其中有半徑、高度、體積以及(高度/半徑)。

要算高度須從面積公式算起，罐頭表面積=桶狀面積+上下兩圓蓋面積，即

$$a=2*pi*r*h+2*pi*r^2, h=(a-2*pi*r^2)/(2*pi*r)=(0.5*a)/(pi*r)-r。$$

體積 $v=pi*r*r*h$ 。

設定 $h=(0.5*a)/(pi*r)-r$ 以及 $v=pi*r*r*h$ ，接著顯示半徑、高度、體積等資料。等待

鍵盤指令。若鍵盤按 n，則檢查需不需要跳頁，若罐頭高度大於 0.2，則將 r 設成 r+i，增加圓半徑，再帶入求 h，v。

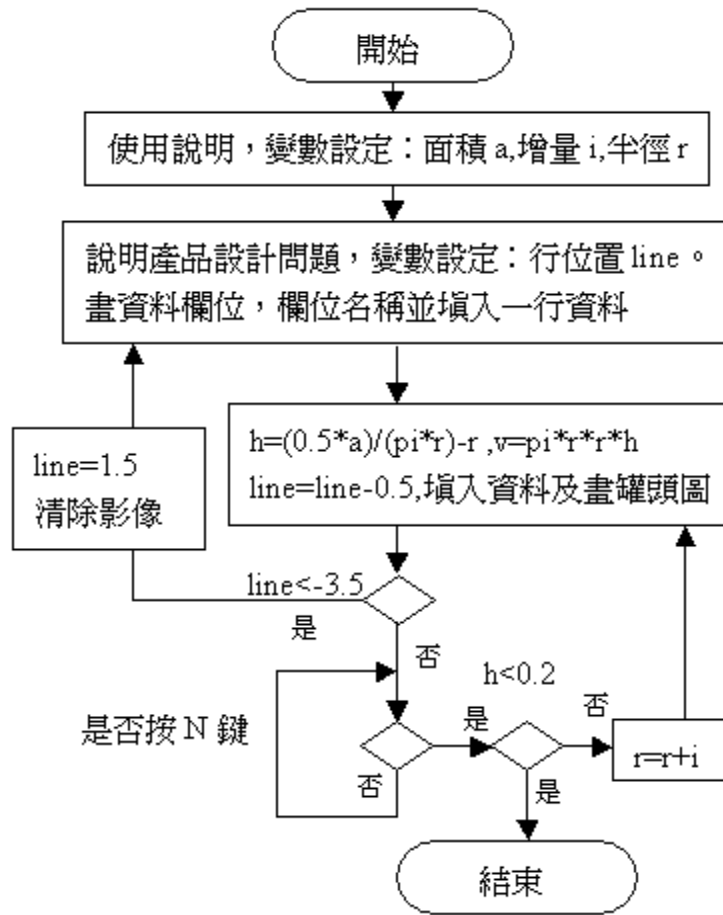


圖 4.11 產品設計程式設計流程圖

三、程式的執行：

首先列印程式資料，程式執行填資料表，並畫出展開的罐頭尺寸圖，如圖 4.12。

小明是大華食品廠的產品設計工程師，有天老闆要他設計一種鳳梨罐頭，罐頭金屬的面積 a=70 平方公分，裡面裝的鳳梨越多越好，小明如何設計？

面積 a 設定範圍：20-80
半徑增量, 最小值：0.01
執行請按 N 鍵得資料。

半徑	高度	體積	高度/半徑
r=1.5	h=5.93	v=41.9	d=3.95

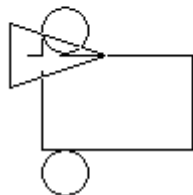


圖 4.12 產品設計程式初執行

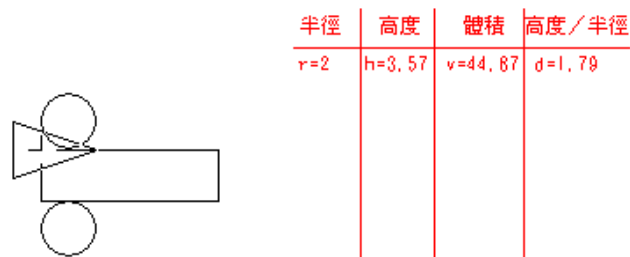
按鍵 n 可填入不同的半徑值再做計算，如圖 4.13 顯示。

半徑	高度	體積	高度／半徑
r=1.5	h=5.93	v=41.9	d=3.95
r=1.55	h=5.64	v=42.55	d=3.64
r=1.6	h=5.36	v=43.13	d=3.35
r=1.65	h=5.1	v=43.64	d=3.09
r=1.7	h=4.85	v=44.07	d=2.85
r=1.75	h=4.62	v=44.41	d=2.64
r=1.8	h=4.39	v=44.68	d=2.44
r=1.85	h=4.17	v=44.86	d=2.26
r=1.9	h=3.96	v=44.95	d=2.09
r=1.95	h=3.76	v=44.96	d=1.93

圖 4.13 產品設計表格放大圖

當顯示資料已經到底，再按一次 n 鍵會清空資料表，進而填入新資料，如圖 4.14。

小明是大華食品廠的產品設計工程師，有天老闆要他設計一種鳳梨罐頭，罐頭金屬的面積 a=70 平方公分，裡面裝的鳳梨越多越好，小明如何設計？



The diagram shows a side view of a cylindrical can with a conical lid. To the right is a table with the following data:

半徑	高度	體積	高度／半徑
r=2	h=3.57	v=44.87	d=1.79

圖 4.14 產品設計程式換頁執行

因為 d 值在 2.09 與 1.93 之間有極值，此時的 r 值為 1.9(d=2.09 時)，減少半徑增量 $i=0.01$ ，顯示體積 v 的精度到小數點下 4 位，讓顯示更精確，修改程式碼如圖 4.15。



The screenshot shows a programming interface with the following settings:

- 將變數: i 設定成: 0.01
- 將變數: r 設定成: 1.9
- 變數: v 精度: 4
- 絕對

Blue arrows point to the input fields for i, r, and the precision setting for v.

圖 4.15 產品設計程式修改

執行結果如圖 4.16：

半徑	高度	體積	高度／半徑
r=1.9	h=3.96	44.9518	d=2.09
r=1.91	h=3.92	44.9598	d=2.05
r=1.92	h=3.88	44.9642	d=2.02
r=1.93	h=3.84	44.9649	d=1.99
r=1.94	h=3.8	44.962	d=1.96
r=1.95	h=3.76	44.9555	d=1.93
r=1.96	h=3.72	44.9453	d=1.9
r=1.97	h=3.69	44.9314	d=1.87

圖 4.16 修改參數後產品設計表格放大圖

可知體積最大值是發生在 $d=2.02$ 與 1.99 之間，亦符合理論上的推測(第 49 頁)。本範例運用算數平均數、幾何平均數所推導的理論與電腦模擬互相印證，學生除了課堂練習外，可以蒐集市面上販售的罐頭尺寸，作一延伸性的探討。教師也可以將市面上販售的罐頭尺寸再一次代入本程式中，做進一步的比較分析。



第四節 牛頓拉夫森法求根號值(程式碼見附錄四)：

一、概述：1690年英國數學家拉夫森(Joseph Raphson, 1648-1715)出版了”Analysis alquationum universalis”，書中描述求平方根近似值的方法，稱之為牛頓拉夫森法。要求某數a開根號的值，則重複使用 $bn = (b+a/b)/2$ ，此處a是被開方的數，b是根號a的原來猜測值，bn是根號a新的近似值。每次我們得一bn值，隨即代入b，得到更新的bn值，一直到b夠接近bn為止，舉例而言，若要求根號98的值，先以猜測值6，代入 $bn = (b+a/b)/2 = (6+98/6)/2 = 11.166666$ ，再將此值設為b，代入bn，當b與bn相當接近時根號98的值即為b。

二、程式設計流程圖：

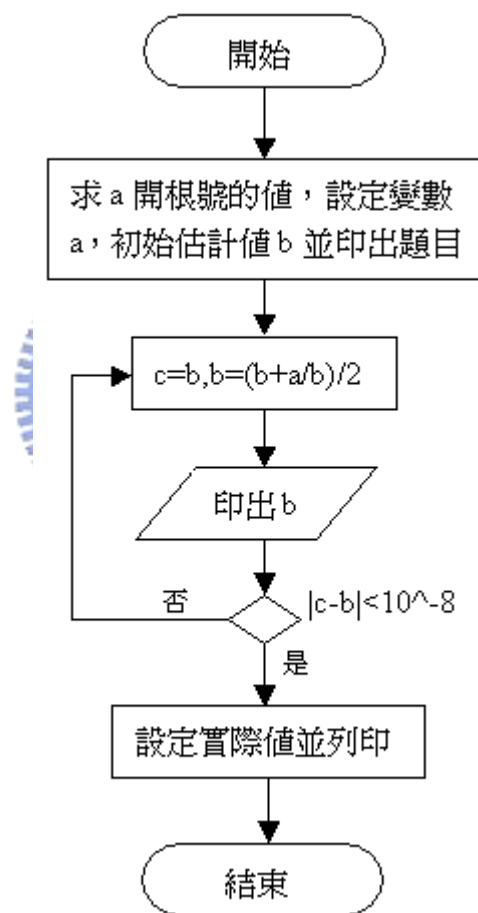


圖4.17 牛頓拉夫森法求根號值程式設計流程圖

三、程式的執行：

```
要算根號 98 的近似值。  
b=11.16666667  
b=9.97139303  
b=9.89975414  
b=9.89949494  
b=9.89949494  
實際值= 9.899494937
```

圖 4.18 牛頓拉夫森法求根號值程式執行



第五節 蒙地卡羅法算面積(程式碼見附錄五)：

一、概述：

蒙地卡羅法是以摩納哥首都蒙地卡羅而命名，在那裡有許多賭博輪盤——一種簡單的隨機數字產生器。蒙地卡羅法是透過亂數以及機率統計來了解問題，本例要以此法來計算圓形面積，進而計算PI值，其原理如下：

圖 4.19 正方形邊長為 4，中間包著一個黑色圓，若要計算圓面積，可隨機在此正方形內抽出 n 點，若有 b 點落在圓內，則由

(落在圓內的點數 b) / (抽取總點數 n) = (圓面積) / (正方形面積)

圓面積 = $(b/n) * (\text{正方形面積})$ ----- (1)

只要知道 n 與 b ，就可以算出圓面積了。

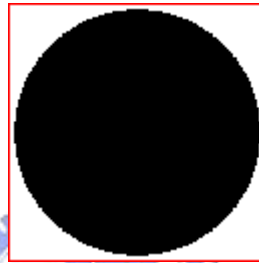


圖 4.19 正方形與圓

Drape 有提供亂數以及顏色檢查的功能，相當適合解決此類問題。

二、程式設計流程圖：

首先畫出一 $4*4$ 的樣本空間，其中畫一個實心黑色圓(半徑為 a)作為代測圖形，接著開始測試。

讓滑鼠移到畫布上任一點，移動直到滑鼠游標落在樣本空間內，判定其下顏色是否為黑色，若為黑色，則變數 b 增加 1，繼續下一次移動，否則 b 不增加繼續下一次移動，執行這段指令反覆 10000 次。

代入(1)式，圓面積 $c = (b/10000) * 16$

又圓面積 $c = \text{圓周率} * a^2$ (a 為半徑)

所以圓周率 $d = c / (a^2)$ 。

因為圓面積是由游標隨意移動而估計出來的，所以每次估計出來的值都不一樣，估計的圓周率值也不一樣。程式的流程圖如圖 4.20。

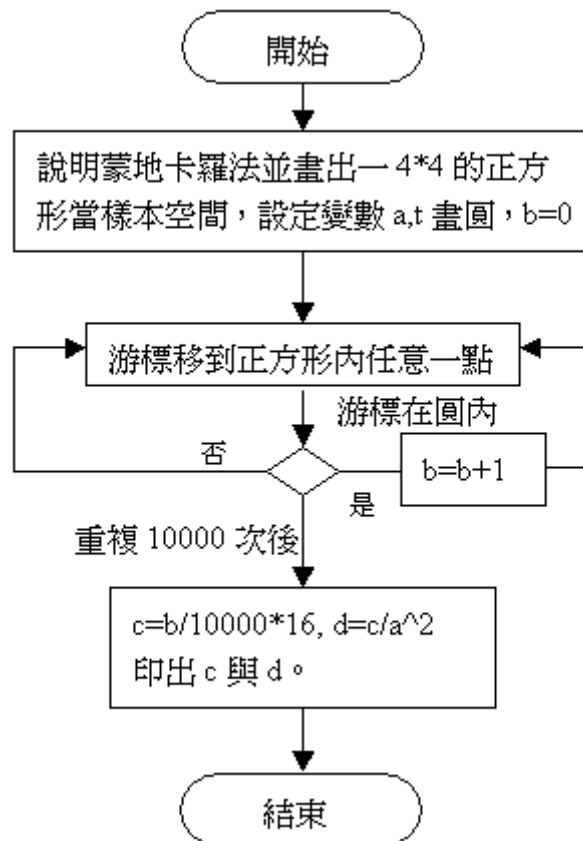


圖4.20 蒙地卡羅法程式設計流程圖

三、程式的執行：

顯示說明畫面，畫一圓形並填滿如圖 4.21。

蒙地卡羅法：在4*4的範圍中，任意選一點並分辨是否在物體內，經過多次選點判別，得到物體概略面積。

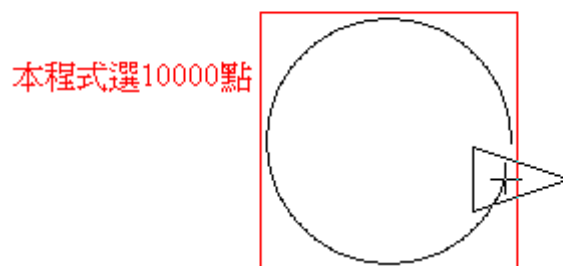


圖 4.21 蒙地卡羅法程式初執行

使游標移到任意處，判別游標下顏色是否是黑色，若是黑色，則 $b=b+1$ 。若非黑色，則游標移到另一任意處。重複以上動作 10000 次，如圖 4.22。

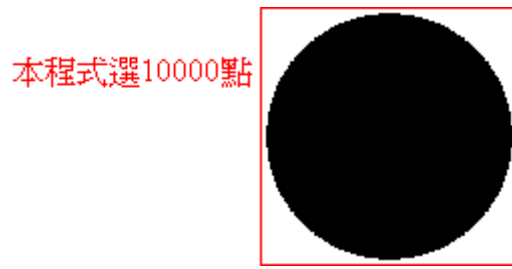
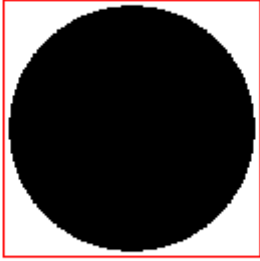


圖 4.22 蒙地卡羅法程式執行中

經過估計，圓面積約為 11.384，PI 值約為 3.153，如圖 4.23。

蒙地卡羅法：在4*4的範圍中，任意選一點並分辨是否在物體內，經過多次選點判別，得到物體概略面積。

本程式選10000點



此物體面積約為：
c=11.384
PI的估計值：
3.153

每次出現答案皆不同

圖 4.23 蒙地卡羅法程式執行結果

本程式範例可以調整游標移動選點數，因為游標移動選點數很多，所以估計 PI 的值相當準，然而如果調整游標移動選 1000 點，則估計 PI 的值會分佈得比較廣，調整游標移動選 100 點，估計 PI 的值會分佈得更廣，所以選擇的次數越多，所求的 PI 值範圍會越小，越接近 PI 的理論值，然而程式執行時間將加長，故本例在執行時間與精確程度的考量下選點次數為 10000 點。

第六節 方格法算面積(程式碼見附錄六)：

一、概述：

使用方格法算圓面積，依然是在正方形中抽出 n 點，統計落在圓內有 b 點(如上例蒙地卡羅法)，就可以算圓面積了，不過在蒙地卡羅法中是隨機取樣，本例卻是將 $4*4$ 的正方形細切為每邊長 0.01 的小正方形(共 160000 個)，游標從正方形的左下角依序掃描到正方形的右上角，判定有多少小正方形在圓內，因將原正方形分格，故稱之為方格法。

二、程式設計流程圖：

首先畫出一 $4*4$ 的樣本空間，其中畫一個實心黑色圓作為代測圖形， $s=400$ ，是分段數。游標移到正方形左下角，往上移 0.01 ，測試若游標下顏色為黑色，則 $b=b+1$ ，往上移 0.01 再測試，若游標下顏色不為黑色，往上移 0.01 再測試，若游標 y 值大於 2 ，則跳往下一列繼續測試，直到正方形右上角為止。此時圓面積 $ans=b*16/(s*s)$ ，圓周率 $c=ans/a^2$ ，流程圖如圖 4.24。

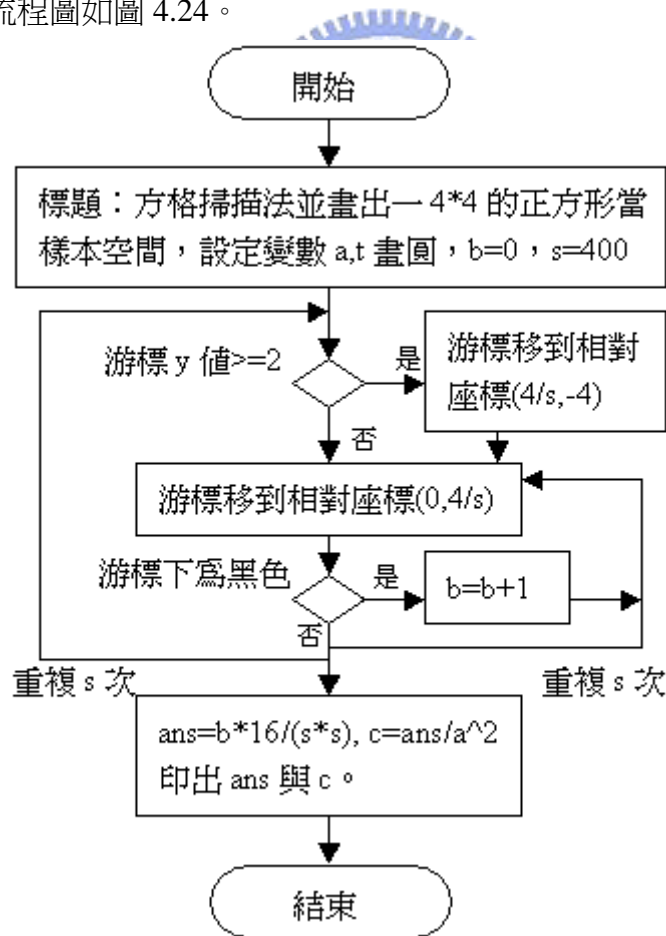
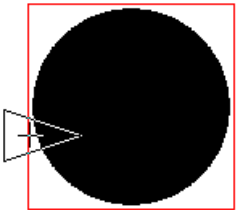
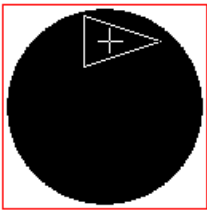
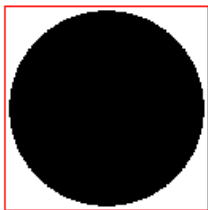


圖 4.24 格法程式設計流程圖

三、程式的執行：

<p style="text-align: center;">方格掃描法</p> 	<p style="text-align: center;">方格掃描法</p> 	<p style="text-align: center;">方格掃描法</p>  <p style="text-align: right; color: red;"> 形狀面積： 11.5373 PI的估計值 3.196 每次出現值相同 </p>
開始執行	執行中	執行後

此處的方格法與上一節的蒙地卡羅法都是在算面積，方格法以類似掃描的方式計算面積，算出的答案是固定的，蒙地卡羅法以機率大小算面積，每次算出的答案不同，除了可以計算規則幾何圖形的面積外，也可以計算不規則圖形的面積。



第五章 教導學生使用及學生作品

本章概述 Drape 的教學實施，並展示教學後的學生作品。第三節說明學生對 Drape 的意見與問卷調查，並針對問卷調查結果提出解釋，在經過兩個學期的試教後，研究者認為 Drape 程式可以用來教導國中學生，認識程式設計。

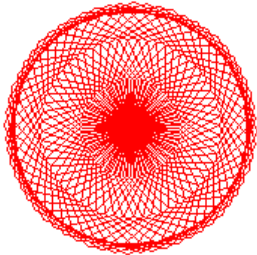

第一節 Drape 的教學

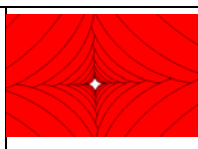

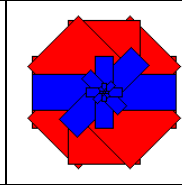

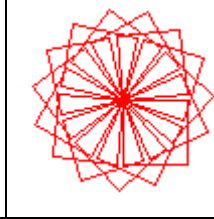


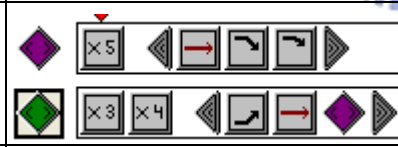
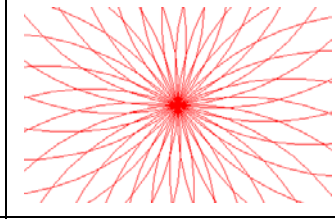
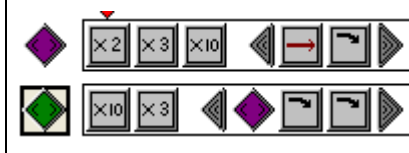
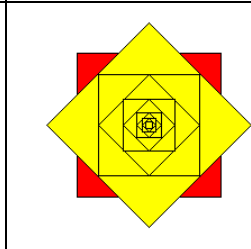
研究者於民國九十二年十一月份，以四個星期，每個星期一節課的時間對國二學生進行 Drape(英文版)教學，第一節課介紹 Drape 繪圖基本指令，第二節課介紹迴圈、程式呼叫、滑鼠指令，第三節課介紹對稱的概念以及三個對稱中心的繪圖模式，第四個星期交一份繪圖作業，要求學生於上課時間完成一個自行創作的圖形。教學以講述法以及示範教學為主，發給學生的講義如附件七至九，並在四個星期後請學生填寫學習意見，作為程式編修的參考，同時進行 Drape 程式的修改，於九十二年十一月將小提示、對話框、錯誤訊息提示以及文字列印加以中文化，九十三年一月新增顯示變數功能，同年二月加入三角函數、乘冪等數學函數，並加入數學運算錯誤時的錯誤訊息提示。民國九十三年四月份，研究者再以四個星期，每個星期一節課的時間對相同的學生做 Drape(中文版)教學，第一節課介紹 Drape 迷宮，第二節課介紹變數計算，第三節課介紹數字排序，第四個星期介紹以 Drape 設計，模擬小畫家的視窗程式—小小畫家，介紹學生如何使用此程式，如何修改程式以設定顏色，並要求學生於上課時間完成一個自行創作的小小畫家作業。發給學生的講義如附件十至十二。

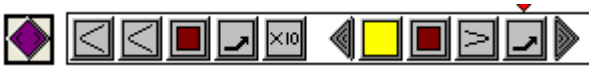
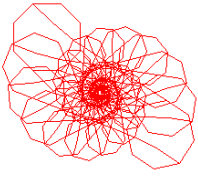

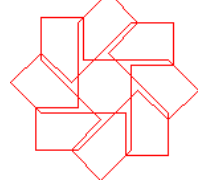

第二節 學生作品

下列學生作品是在上學期學生上完四週 Drape 的課程之後學生自己設計的作品，研究者發現學生已經學會基本的繪圖指令、重複指令以及呼叫程序指令，經研究者評閱後選出代表性的學生的作品及程式碼如表 5.1：

表 5.1 學生使用 Drape 指令所畫出圖形

例一	
程式碼	

例二	
程式碼	
例三	
程式碼	
例四	
程式碼	
例五	
程式碼	
例六	
程式碼	
例七	

程式碼	
例九	
程式碼	
例十	
程式碼	

研究者上課時曾提到對稱的圖形符合美感標準，並先給予學生三種對稱造型程式(附錄十二)，大部分學生會修改程式而得不一樣的對稱圖形，也有學生以嘗試錯誤的方式自行繪製圖形，研究者也發現部分學生對角度旋轉認識不足，無法在下指令時預測程式畫出的圖形。

下列學生作品是在下學期學生使用小小畫家程式而設計的作品，研究者希望學生能夠看得懂設計此程式的原 Drape 程式，並能夠修改程式，經研究者評閱後以圖案的美感呈現，選出代表性的學生的作品，如表 5.2：

表 5.2 學生用小小畫家完成之作品

學生有能力修改原 Drape 程式的背景以及選擇顏色設定，如表 5.2 第二行的學生作品。學生看得懂 Drape 程式指令，並加以修改。

第三節 學生意見

於民國九十二年十二月份，研究者設計問卷讓學生填寫學習意見，總計回收問卷 110 份，其中有填答 104 人，表達喜歡或不喜歡 Drape 的理由，6 人無任何意見。統計結果如表 5.3。

表 5.3 學生學習 Drape 意見統計表

編號	喜歡 Drape 的理由	人數	編號	不喜歡 Drape 的理由	人數
1	可寫程式	1	1	無中文介面	8
2	畫圖功能強大	82	2	畫面太小	8
3	方便迅速	14	3	顏色太少	1
4	介面佳	3	4	色調太暗	2
5	功能多	1	5	指令複雜	4
6	不當機	1	6	無法畫立體圖	1
7	好玩	2	7	不能畫可愛的圖形	14
			8	功能少	3
			9	要刪除指令很辛苦	8
			10	速度慢	1
			11	常當機	3
			12	要思考很麻煩	4
			13	麻煩無聊	21

有 104 人提到喜歡 Drape 的理由，學生大多認同 Drape 的繪圖功能，在不喜歡的原因方面，編號 1 到 5 與介面有關，編號 6 到 11 與系統功能有關，編號 12、13 與使用者有關。編號 1 無中文介面促使研究者將 Drape 中文化；編號 2、3 畫面太小與顏色太少是學生還不熟悉 Drape 指令而誤寫的意見，不予處理；編號 4 色調太暗乃因 Drape 沿用 Windows95/98 的色系，尊重原創作擬不修訂色調；編號 5 指令複雜提醒研究者讓操作介面儘量單純；編號 6 畫立體圖而言，Drape 是可以畫一些簡單靜態的立體圖，動態立體圖因涉及觀察者位置，相當複雜，因此暫不考慮；編號 7、8 不能畫可愛的圖形與功能少，促使研究者思索視窗程式的可行性並開發小小畫家程式；編號 9 到 11 是學生還不熟悉 Drape 指令而誤寫的意見，其中開新檔案就可將指令完全清空，執行速度可用快速執行調整，當機的發生多為程式未終止產生，故不加以處理；編號 12、13 是操作者主觀意見的陳述，其中 21 人覺得麻煩無聊，促使研究者對此軟體新應用做進一步的探究。

Drape 程式經研究者修改並做第二次試教後，於民國九十三年六月份，讓學生填寫另一份調查問卷，問卷調查的題目如附錄十六，經學生 111 人填答(多一轉學生)，填答

結果分析如表 5.4 及圖 5.1：

表 5.4 問卷調查統計表

問卷調查題目	非常同意	同意	無意見	不同意	非常不同意
1. 我覺得學習 Drape 可以幫助我了解程式設計的概念。	14	40	43	9	5
2. 我覺得學習 Drape 對我而言很困難。	13	28	42	23	5
3. 我覺得學習 Drape 可以訓練邏輯思考能力。	20	39	37	11	4
4. 我覺得 Drape 是一個很有特色的軟體。	16	28	51	11	5
5. 我覺得 Drape 畫出的圖形很呆板無聊。	20	20	45	21	5
6. 學習 Drape 讓我對電腦更有興趣。	8	15	45	30	13
7. 當我下 Drape 指令時我可以了解程式執行之後的結果。	13	31	42	17	8
8. 學習 Drape 可以激發我的創意。	13	33	38	20	7
9. 我對自己用 Drape 設計出來的圖案(作品)很滿意。	10	22	43	22	14
10. 整體而言，我喜歡 Drape。	3	14	57	22	15

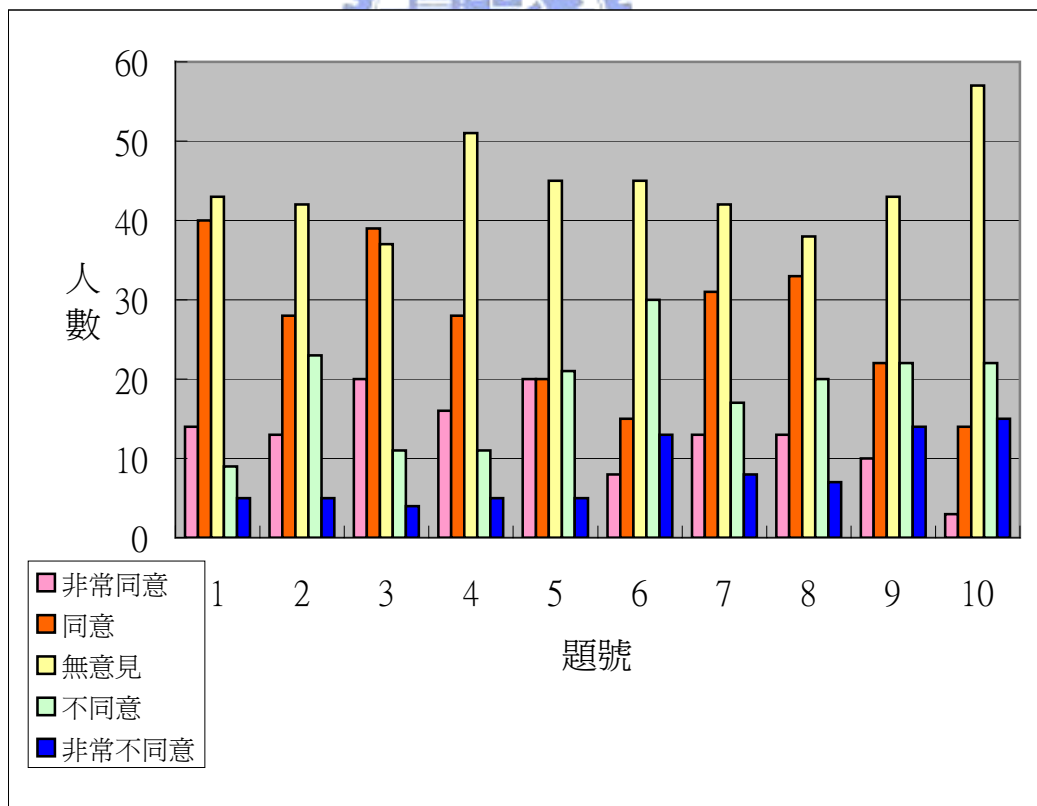


圖 5.1 問卷調查題號與人數統計直條圖

在每一個選項中以無意見最多，也許五點量表不適合學生作答，未來研究填答類似問卷時宜把無意見拿掉，但其他選項經整理後可以得到學生的觀點如下：Drape 是很有特色的軟體、可以幫助了解程式設計的概念、訓練邏輯思考，在執行前就知道執行結果、可以激發創意；另一方面而言，Drape 是困難的、畫出圖形是呆板的、不會增加對電腦的興趣、對設計出來的圖案不滿意且不喜歡 Drape。

研究者嘗試解釋學生的觀點：對於學生而言，Drape 的繪圖是麻煩的，因為一切從無到有都要自己動手，Drape 設計出來的圖形是呆板的，不如學生常接觸的 3D 立體動畫，與動畫相較，對 Drape 作品也不會滿意，程式設計本質上是辛苦麻煩的，除了特定的幾個學生之外，學習 Drape 程式設計不會增加對電腦的興趣，也不會喜歡 Drape。


另外、研究者發現學生在使用 Drape 繪圖時幾乎沒有任何問題，都可以在時間之內將圖形畫好，然而在介紹變數的輸入以及變數的處理之時，對學生造成了大問題，他們只會做簡單的設定如將 a 設定為 $3*2$ 等等，而將 a 設定為 $a+1$ 的意思有一大半同學不了解，這是否和學生的變數概念了解有關，值得進一步研究。以至於在使用變數設定寫程式時，大部分同學都寫不出來(一班約只有 2 人做出)，學生沒有程式設計的基礎，以致於覺得程式設計既麻煩、又困難。

有 59 人(53%)認為學習 Drape 可以訓練邏輯思考能力，有 54 人(48%)認為學習 Drape 可以幫助了解程式設計的概念，大體而言學生對此軟體是抱持著肯定的態度的，研究者亦認為 Drape 相當適合學生認識程式設計。

第六章 結論

本章對本研究作一總結，內容包括：研究結果、研究建議、以及 Drape 程式未來發展方向研究者也提出以 Drape 程式教導特殊學生，以 Drape 程式設計造型，以及軟體功能的再加強，當作 Drape 程式未來發展方向的參考。

第一節 研究結果

本論文研究目的有四：一、探討 Drape 程式在國中施教的可行性。二、Drape 程式中文化。三、延伸 Drape 程式基本功能的探究。四、以 Drape 程式開發解決問題之應用程式範例。在探討 Drape 程式在國中施教的可行性方面：經由兩學期的試教，學生皆能在課堂上完成指定作業，且此軟體獲得多數學生肯定，對學生程式設計的觀念有幫助，可於國中階段實施教學。在 Drape 程式中文化方面：已經將小提示、對話框、錯誤訊息提示以及文字列印加以中文化。在延伸 Drape 程式基本功能的探究方面：新增三角函數 \sin 、 \cos 、反三角函數 \arctan 、冪次方以及取整數功能並新增顯示變數  功能。在開發 Drape 程式解決問題之應用程式範例方面：撰寫應用程式，計有韓信點兵、聯立方程、產品設計、牛頓拉夫森法、蒙地卡羅法、方格法等，作為 Drape 程式應用之初探。

第二節 研究建議

根據以上研究結果，研究者提出兩點建議：

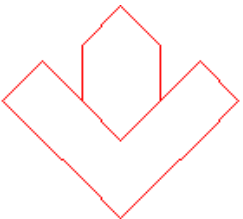
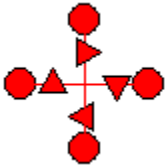
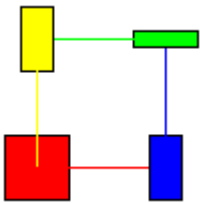
- 一、Drape 可以作為認識程式設計之入門軟體：依照九年一貫課程七到九年級資訊教育學習內涵所規定，中學生需學習程式設計，經研究者開發編修後，目前 Drape 是已中文化的、簡單、功能強大、層次分明、對硬體架構與軟體系統要求不高、穩定而且免費。故可作為中學生學習程式設計之軟體。
- 二、Drape 可以作為學習數學之輔助工具：它可以畫出數學函數、可以用數學陳述來計算算式，有重複執行程式的能力來計算數值問題、有亂數功能以模擬問題，可作為數學解題之輔助工具。

第三節 Drape 程式未來發展方向

研究者將從題材的製作運用以及軟體功能的再加強兩個角度探討 Drape 未來的發展方向。在題材的製作運用上，研究者認為：

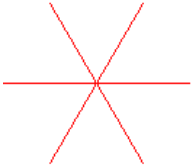
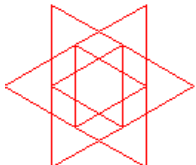
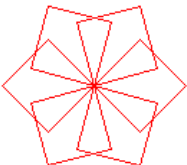
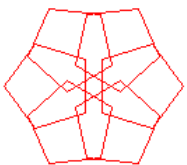
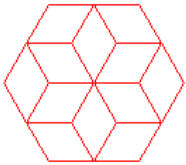
- 一、Drape 或許可以用來教導特殊學生：上電腦課時，有些平常無法繳交指定作業的資源班學生，並不需要老師特別在旁指導，也能毫無困難地使用 Drape 的基本繪圖指令。在任教的班級中有三位情形比較嚴重，甚至不會用注音拼自己名字的學生，可以使用 Drape 畫出如表 6.1 的作品，顯示這些學生有能力畫有意義的圖形。研究者認為，或許經由適當的課程設計，Drape 可以用來教導特殊學生，並增進其數、量、形的觀念。

表 6.1 特殊學生使用 Drape 指令畫出的作品

		
作品 1	作品 2	作品 3

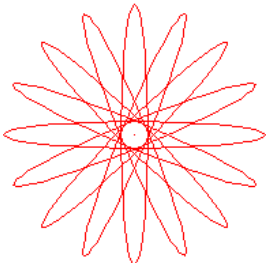
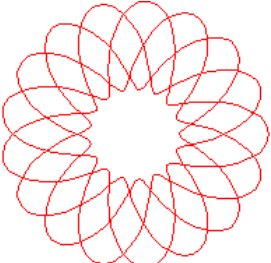
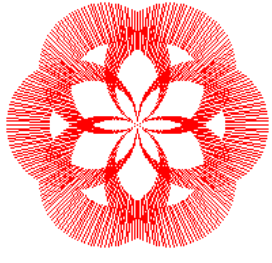
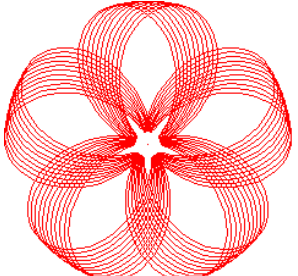
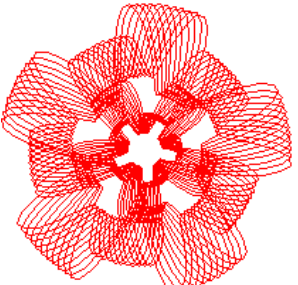
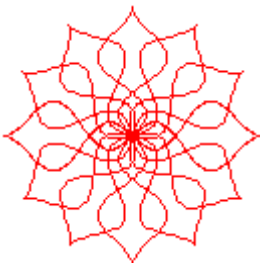
二、Drape 可以用來設計視覺造型：Drape 很方便繪製一些簡單的幾何圖形構圖。例如，可寫一適當程式，製造 6 個正 n 邊形，每次繞原點旋轉 60 度，以繪製對稱原點之圖形，調整 n 的數目，可以得到如表 6.2 之造型：

表 6.2 Drape 造形設計結果一例

				
n=2	n=3	n=4	n=5	n=6

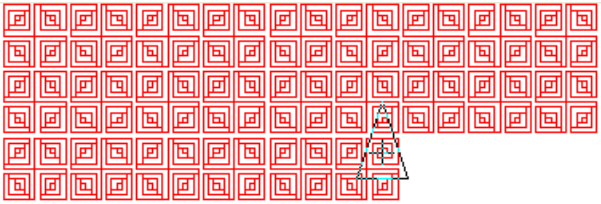
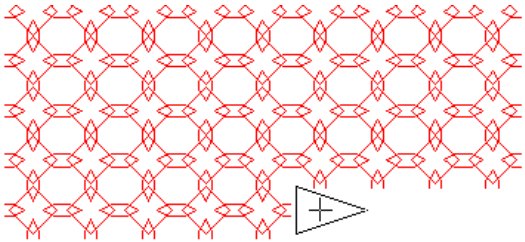
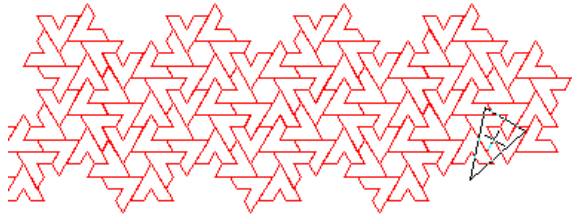
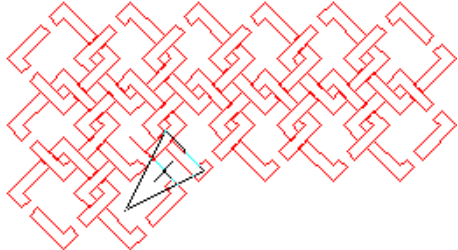
又如，增加三角函數的功能可繪製一些『漂亮』圖形如表 6.3：

表 6.3 使用三角函數畫出之圖形

其他如，Drape 也可以設計類似鑲嵌的圖案如表 6.4：

表 6.4 一些鑲嵌的圖案

	
<p>類似地毯的圖形</p>	<p>類似瓷磚的圖形</p>
	
<p>鋸狀物的集合</p>	<p>鍊子的集合</p>

三、在軟體功能的再加強方面，研究者認為還有改進空間，以下提出四點未來研究方向：

1. 數學函數的增加：有哪些數學函數是需要再增加的？增加後有沒有新的應用？
2. 運算精度的增加：在 Drape 中的數值計算可以精確到何種程度？如何滿足電腦數值計算的要求？
3. 改變對話框的可行性：有些程式執行前需改變變數的初始值。現行改變變數的方法是在拖曳程式區的變數設定圖示上按滑鼠右鍵，開啓對話框後輸入新數值，如果程式中有兩個以上的變數設定圖示時，可能會按錯圖示，到開啓對話框後才發現。比較好的做法是在程式執行時自動帶出要改變的變數對話框，等待輸入數值之後再繼續執行程式。改變對話框的標準作業方法尚待研究，又若新增此功能，原程式碼可能要大幅修改，故留待下階段再針對此點作改進。
4. 陣列變數的增加：增加陣列變數可以使變數輸入更有效率，也可深入探討數學矩陣運作，故在許多軟體（如 BASIC、FORTRAN 等）中皆有陣列變數。但陣列變數的輸入、輸出、維度限制與運作尚待研究，留待下階段再針對此點作改進。

上述功能的新增與改變必須在一個整體的觀念下運作，希望所做的修改是易於操作，可以理解的。Drape 是一個年輕的軟體，它的可塑性相當大，也一定有可改進的空間。本文以 Drape 為中心，紀錄研究者試用、試教、發現、改良、改進的過程，雖然程式編修以及應用尚有若干缺點，然而它的優點是遠遠多於缺點的。研究者在研究過程中獲得不少啓發，且愉快地進行每一項研究，希望新編修過的 Drape 也能如此嘉惠予每個學習它的人。