

第二章 文獻探討

本章第一節是探討初學者學習程式設計的困難，第二節介紹 Lego Mindstorm 教學工具在教育及電腦科學上的應用，第三節是討論 Lego Mindstors 如何應用在程式設計教學。

第一節 程式設計初學者所遇見的困難

以下依 du Boulay (1989) 就初學者學習歷程中所舉出五個容易造成學習困難的範疇分別予以討論。

一、 程式設計的基本概念、定義

如程式設計是什麼？設計完成的程式可以用來完成怎樣的工作？還有實務上的技巧，如程式測試、除錯、撰寫等。

二、 程式語言的語法及其含義、用法

如何正確地瞭解程式語言的語法及其含義、用法是初學者常遇到的另一種困難。由於有些程式語言指令使用字詞與自然語言相同，如 if、for、while 等，初學者常會用自然語言的意義與程式語言指令的意義作不當類比，他們會根據自然語言中 while 的含義，推想程式語言 while 的條件式是一直持續執行，而不是每執行一次迴圈條件式需再作一次的檢驗 (Bonar & Soloway, 1989)

三、 程式結構知識的了解

程式結構知識的了解，所謂的知識，包括了三方面的知識，首先是在設計方面的知識，即是知道設計程式的方法，如由上而下、結構化等，其次是在實作方面的知識，知道程式語言的語法，了解如何運用程式語言的程式庫，或是使用程

式語言的整合環境的各項工具，最後是在程式的除錯方面，除了明白該如何除錯，對除錯工具的使用也有一定程度的了解（Robin, Rountree & Rountree, N. 2003）。

四、 問題策略規劃能力

問題策略規劃能力，策略即是指運用相關的知識（程式結構）在各個相對應的範疇，往往這是一個關鍵的地方，因為在現今許多的程式設計的課程都著重在介紹程式語言的語法結構，課程的安排也著重在如何呈現這方面的知識，但是這種的安排，往往是忽略了知識應如何運用在程式設計中。因此有許多的課程設計，除了介紹有關程式設計的基本知識，亦相當著重在程式設計的策略面，即著重在分析程式設計者在寫、閱讀程式所運用的方法、策略（Davies, 1993）。

五、 抽象表徵機器模型（Notional Machine）的建立

抽象表徵機器模型（Notional Machine）的建立，這是一個心智模式用以說明執行中程式的行為，使初學者能夠瞭解他所撰寫的程式在電腦中所發生的事。許多研究（Canas, Bajo & Gonzalvo, 1994; du Boulay, O'Shea & Monk, 1989; Hoc, & Nguyen-Xuan, 1990; Mayer, 1989; Mendelsohn, Green & Brna, 1990）均特別強調心智模式的重要性。就初學者而言，一個有效的機器模型應是簡單、能輔以具體工具來呈現程式執行過程中所發生的事件（du Boulay, 1989），也可讓他們運程式語言來操控這工具，透過這樣的過程他們發現程式運作，了解正在執行中程式的行為、及其內裏所隱藏的細節。

程式設計初學者所遇見這五類的困難，彼此之間並不是一種先後、順序發生的關係，而是會在初學者學習過程中混合穿插出現或是同時出現數個困難，因此

「迷你程式語言」的提倡 (Brusilovsky, Calabrese, Hvorecky & Kouchnirenko, 1997), 使用一個指令少、語法結構簡單的程式語言, 以減少學生學習程式設計時程式語言語法的認知負荷。另外一方面, 使用圖形化或是使用實體的機器人以呈現程式結果, 讓學生能觀察執行中程式的歷程, 逐步建立一個正確的抽象表徵機器模型, 提昇學生解題能力, 降低學生初學程式設計的困難。

第二節 Lego Mindstorms 的軟硬體

在本節中將會依序介紹 Lego Mindstorms 的硬體構成主要元件、操作模式，其次是討論 Lego Mindstorms 的程式設計工具。

壹、Lego Mindstorms 的簡介

LOGO 是美國麻省理工學院 Papert 教授與其研究伙伴於 1960 年代發展出來的一種程式語言，目的在提供一種簡易且功能強大的電腦程式語言，以幫助兒童學習使用電腦，進而促進兒童的問題解決能力。雖然 LOGO 在教育界造成轟動，享有許多好評，但曾師事皮亞傑（J. Piaget），對皮亞傑理論知之甚深的 Papert 教授仍然覺得有一個遺憾，就是 LOGO 程式語言裡的小海龜只能停留在螢幕上，對許多認知層次仍屬具體運思期（Concrete Operational Stage），需要藉實物進行思考的小學生而言，仍有不盡完善之處（田耐青，1999）。於是「如何讓小海龜躍出螢幕，讓孩子不僅看得到，也摸得到？」就成為 Papert 的下一個奮鬥目標。Lego Mindstorm 即是源自於 Papert 和他的實驗團隊 Lego/Logo 的研究計劃（Martin, 1994），在這計劃中他們原想使用 Logo 語言以控制 Lego 公司的一個產品 Lego Technic，這產品除了一般的 Lego 積木之外，還包括各種新的零件如齒輪、輪子、馬達等，Lego 公司在獲悉這項計劃後，與 Papert 實驗團隊討論相關細節，促成了 Lego 公司和 Papert 實驗團隊的合作發展 Lego/Logo 計劃。

Lego Mindstorms 的硬體有兩個主要部分，一是用來組裝各種模型的基本積木元件，另一個則是可程式化的積木。Lego 積木幾乎是每個孩童都玩過的益智玩

具，有許多不同顏色，不同尺寸，呈長方體、正方體或其他立體形狀的塑膠塊，一面呈溝狀，另一面有多個圓柱形的突出物，可以讓塑膠塊適度的咬合在一起；除了這些基本的塑膠塊之外，Lego Mindstorms 也提供齒輪、皮帶、連接用的插銷。而可程式化積木在 Lego Minstorm 產品系列稱這為 RCX (Robotics Command Explorer <http://www.lego.com/eng/education/mindstorms/default.asp>)。RCX 是一個 Lego MindStorms 產品系列的控制中心，它的外觀與一般的 Lego 積木相似 (圖 2-1)。RCX 使用六顆 3 號乾電池。液晶面板的設計，可以清楚看到程式執行時，感應器的反應數值。前端的紅外線接收器，可同時接收電腦的指令，或是與另一個 RCX 相互傳輸。RCX 具備三個輸入接點，一般皆標示為 1、2、3 可連接光源感應器、觸控感應器、溫度量測器等，另外有三個輸出接點，標示為 A、B、C 可連接馬達及燈泡等積木。RCX 可以接受電腦程式的控制，做出各種不同功能的動作，電腦程式傳輸至 RCX 的方式是透過紅外線程式傳輸器 (Infrared Transmitter Tower，簡稱 IR Tower)，紅外線程式傳輸器則透過 USB 或 COM 通訊埠與電腦連接 (Barnes, 2002)。



圖 2-1 可程式化積木 RCX 的外觀及輸出、輸入設備。

圖 2-2 是建構使用 Lego 機器人的完整過程，這並不是一個單一直線的過程，而是一個循環、階梯式的過程，往往有可能在下一輪的設計中，放棄原有想法，而採用新的創作方式（Wang, 2004）以下就各部分作一簡單的描述：

1. 設計與建造

每一個機器人的建造，都有其特定的目的，如要設計一個機器人，能定時定點記錄室內的亮度，和一個機器人能記錄室內各處的亮度，雖然要求似乎很相似，但其設計、建造卻是大不相同。

2. 在電腦上寫程式

運用 Lego Mindstorms 的操控軟體，配合上一個步驟所建造的機器人，設計機

器人的運作程序。

3. 下載程式到 RCX 上

將設計完成的程式，透過紅外線傳輸器（IR Tower），傳送至可程式化控制積木 RCX。

4. 執行程式與調整、測試

執行程式，並觀察機器人的動作是否按照原有設定的程序動作而且正確地執行，或其動作只有部分符合原有設定的情景，此時即需找出程式錯誤的地方，修改後重新下載，再執行直至無誤。

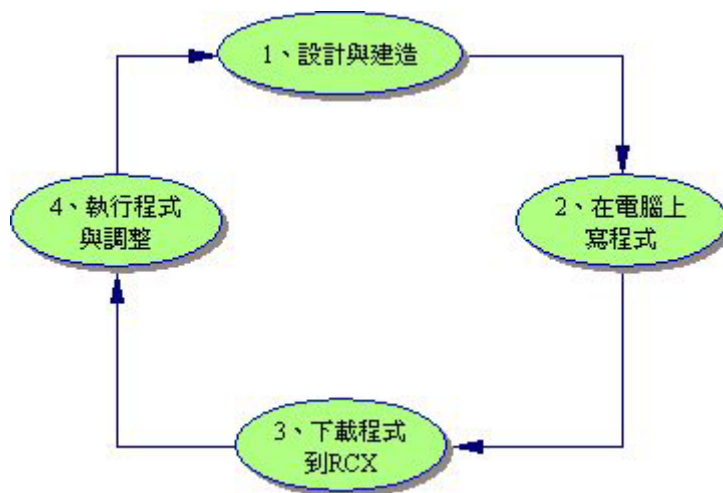


圖 2-3 Lego Mindstorms 的操作模式

貳、 Lego Mindstorms 的程式設計工具

Lego Mindstorms 機器人與 Lego 公司其他機器人不同之處，在於 Mindstorms 系列的機器人，由於有 RCX 作為機器人的「大腦」可以接受使用者的指令而有相對的運作，換句話說，使用者可以透過所撰寫的程式，命令機器人按照他設計的方式去運作。使用者所使用的 Lego Mindstorms 控制軟體可分為三大類(Klassner & Anderson, 2003)，第一類是 Lego 公司本身所提供的程式語言，在這類別因著產品銷售對象，又可分兩種，RIS (Robotics Invention System) 和 ROBOLAB，它們都是以一個小圖示代表某種功能，一個控制的程式即是將這些小圖示用某種的合乎邏輯方式(演算法) 加以組合。第二類仍然使用 Lego 公司所提供軟體作為基準，但不使用 RIS、ROBOLAB 這類程式作為使用者的操作軟體，而是使用一般的程式語言，如 C、JAVA 等。第三類則是更換 Lego 公司所使用的軟體，由於軟體有許多不同的形式，如以 C、C++ 為程式語言的 LEGOS、以 FORTH 為程式語言的 PBFORTH 以 JAVA 為程式語言的 LEGO JAVA OPERATING SYSTEM(簡稱 LeJOS)，以下將分別就這三大類的控制軟體作簡要說明，各操作細節請詳參考文獻中產品手冊、網站。

一、 Lego 公司所提供的程式語言

Lego 公司在推出 Mindstorms 系列的產品，由於市場區隔的考慮，因此推出了 Lego Mindstorms Robotics Invention System 和 Mindstorms for Schools 兩種不同產品組合，就硬體而言，兩者均以 RCX 可程式化積木搭配不同數量的樂高積木、感應器，隨著這些零件不同，所能組合的機器人外觀會有所不同，但原則上這兩者的差異並不大，差別是在控制軟體，這兩個軟體均是運用圖示作為控制程式的

基本元素，但在功能上卻不盡相同（Barnes, 2002; Klassner, 2003）。

1. Robotics Inventrion System (RIS)

由於 RIS (<http://mindstorms.lego.com/eng/products/ris/index.asp>) 銷售對象針對一般的青少年、社會大眾，所選用的軟體也以操作簡易、功能簡單的 RCX code 為主，RCX code 是 Lego 公司在 RIS 套件中所提供用以控制 RCX 機器人的程式語言工具，RCX code 的主要特點就是容易使用，這程式語言是一個完全以圖形、事件導向的環境，在這環境中並不需要寫任何的程式碼，所要作的只是用滑鼠將各個程式方塊，以有意義且合乎邏輯的方式，將這些程式方塊堆疊在一起，再設定各個參數，即可驅使機器人按照指令運作。這個過程與樂高積木的堆疊相似，只是現在所使用的每一個積木是一個個含有不同功能的程式方塊，如馬達的動作、感應器狀態等。另外為了方便操作，它也提供類似副程式的方塊組合，稱為 BIG BLOCK。

2. MindStorms for School

Mindstorms for school (<http://www.lego.com/eng/education/mindstorms/default.asp>) 是針對學校的教育用途，所使用的是控制軟體是功能較多元的 ROBOLAB，ROBOLAB 是 LEGO DACTA MINDSTORMS 或 MINDSTORMS FOR SCHOOLS 系列產品中所附的操控軟體，ROBOLAB 是基於由 National Instruments 所發展的 LabVIEW 圖形化介面軟體 (Wang, 2004)，和 RCX-CODE 一樣，完全圖形化的操作界面，不需要記憶任何的指令，只要利用滑鼠輕鬆的點選圖示，這些圖示代表不同的控制程序，再將這些圖示串連起來，就可以完成複雜的程式設計(圖 2-3)，功能上 ROBOLAB 比 RCX-CODE 多元，ROBOLAB 可以使用變數、多工、資料記錄等，使得以 ROBOLAB 為控制軟體的樂高機器人的應用比較多樣化。

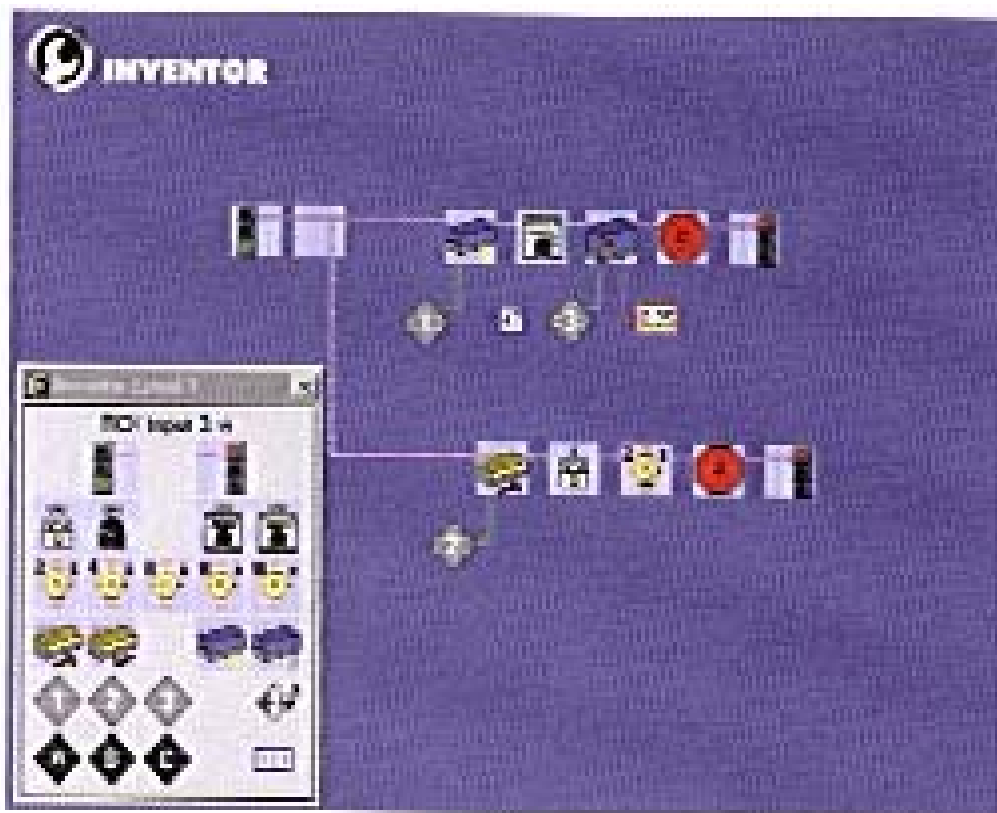


圖 2-3 RoboLab 的範例程式

二、 使用一般的程式語言

在第二類、第三類的 Lego Mindstorms 程式語言比較多元，這是由於 RCX CODE、ROBOLAB 雖然可以讓初學者很快學會操控機器人，也能學習相關邏輯順序相關觀念，但程式比較複雜時，就會使得程式不易閱讀，因著 Prouddfoot 將 Lego Mindstorms 內部的結構分解(<http://graphics.stanford.edu/~kekoa/rcx/>), 在網路上公佈其內部的相關指令，使得其他愛好者能發展其他控制軟體。這兩類的差別只在於是否有使 Lego 公司所使用的韌體，以下分別作敘述。

1. 應用 Lego 公司的韌體

沿用 LEGO 公司的韌體 RCX，但不使用 RCX CODE、ROBOLAB，使用標準的程式語言如 JAVA、C。在這其中，較多使用的是 NQC (Not Quite C: <http://bricxcc.sourceforge.net/nqc/>) NQC 是一個使用在 Lego MindstormsS 的程式語言，它的語法結構近似 C 程式語言，因此已熟悉 C 程式語言可以很輕易地使用這語言來操控 Lego Mindstorms。另一個是 RCXPort 與 NQC 相似，但使用標準的 JAVA 程式語言，透過標準的 JAVA COMMUNICATION API 操控 Lego Mindstorms. (RCXPort: <http://www.slewis.com/rcxport>)

2. 非 Lego 公司的韌體

在上一種模式，使用一般的程式語言作為操控 Lego Mindstorms，但仍是透過 Lego 所使用的韌體作為提供服務的基礎，猶如一般在 PC 上的應用程式所需要的服務是透過作業系統獲得。若想進一步控制 LEGO 的硬體資源通常是更換韌體，常用來更換 RCX 的韌體有以 C/C++ 為主的 LEGOS(<http://brickos.sourceforge.net>), 或是以 JAVA 為基礎的 LeJOS (<http://lejos.sourceforge.net>), 至於其他的，則有

以 FORTH 程式語言寫成的 PBFORTH。

這三者除了程式語言不同之外，LEGOS 在控制硬體的資源最有彈性，讓程式設計者完全掌控 Lego 的各個元件，LeJOS 則提供一個完全支援標準的 JAVA 程式語言的環境，而 FORTH 是一個發展於 60 年代，主要使用機器人、自動控制的程式語言，Ralph Hempel 所發展可下載至 Lego 的版本，可以使用終端機的命令模式作控制。

LeJOS (Lego Java Operating System) 系統類似 Java 的 Micro Edition，為了讓程式庫能更精簡，沒有提供標準的 Java 語言具有的功能，如記憶體自動回收機制 (Garbage collection)、Switch 控制流程指令、64 位元整數型態 (Long variable) 的數學運算等。

在 LeJOS 的 API 除了含有 Java 程式語言的標準功能，還包括控制 Lego Mindstorm 各個元件的函式，可分為下列這幾大項：

- 1) RCX 面板按鍵 (Button)。
- 2) RCX 顯示面板 (LCD)。
- 3) 馬達 (Motor)。
- 4) 感應器包括觸感、光感、溫度、轉速感應器 (Sensor)。
- 5) 紅外線傳輸 (Infrared Interface)。

總之有多種的各具特色的程式語言可以操控 Lego Mindstorms 的運作，但由於本實驗希望學生在習得程式設計的相關觀念之時，對一般的程式語言亦有初步的認識。在本節中所討論的程式語言，LeJOS 提供了完全支援標準 Java 的程語言的環境，而所提供用以操控 Lego Mindstorms 各元件的副程式也相當完備，因此在本研究中採用 Java 程式語言並用 LeJOS 更換 Lego Mindstorms 的韌體。

第三節 Lego MindStorms 與程式設計教學

依照 Lego Mindstorms 的操作模式，在實際教學的運用上，大致可分為兩大類，一是學生需完成整個設計程序，即依據題目要求設計、組裝機器人，然後撰寫操控程式，另一種是著重在操控機器人的程式設計部分，機器人可能是由教師先行組合提供給學生使用或教師提供學生組裝機器人的步驟。以下分別就這兩方面作文獻探討。

壹、完整的機器人設計模式

Flowers & Gossett (2002) 為學生設計一個機器人操作情境的作業，並且詳細列舉機器人的功能、規格，如機器人從什麼地方開始它的任務，在過程中它必須遵循怎樣的規則，怎樣的狀況之下它才算是完成任務等。學生從作業的需求中作問題分析，以決定要完成任務的各項小目標，輸出、輸入條件，各種假設、限制條件，以及在解決問題所需要的公式，並且學生要彙整這些分析成為一份需求分析。然後學生根據這些要求，設計、組裝合適的機器人和設計程式讓機器人能完成任務。在這樣解決問題中，程式設計的基本觀念如循序、條件判斷、迴圈等已經融入在作業的情境中，學生完成作業時也能加強程式設計的相關概念的了解。

Wolz (2001) 和 Gracia & Mc-Neil (2002) 也使用類似的教學方式，提供學生一個機器人的學習情境，學生要在限定的時間內完成並提出一份設計的需求分析文件，學生需要先擬定計劃，計算專題完成所需要的時間、為了符合題目的情境要求程式上的所需的函數量、所需要的人力等，並且完成相關的專題計劃文件，簡而言之，他們期待透過這樣的過程讓學生能了解軟體工程的歷程。

另一個典型運用 Lego Mindstorms 的例子，是將其運用在人工智慧的課程中，Klassner (2002) 讓學生設計從最簡單的機器人 (simple-reflex robot design) 到複雜的機器人如能抓球的機器人、8-Puzzle solver 等，學生在觀察機器人的運作當中，了解人工智慧課程中的一些比較抽象的演算法。而 Kumar, (2001) 則運用 Lego Mindstorms 機器人說明人工智慧中的搜尋和專家系統，如設計一個走迷宮的機器人，這機器人運作的方式是依據探索式的搜尋中的爬山方法和最佳優先搜尋法。機器人必須能感應到迷宮中的分隔壁，當找到出口時機器人要發出一響聲。在測試的時候，學生可以選擇機器人進入的方向，教師則選不同的方向，藉此以證明學生所設計的機器人是適用不同的環境。Kumar 指出學生認為使用一個實體機器人比較困難的地方是如何讓機器人正常的運作，就學習興趣則全部的學生都希望能繼續使用 Lego 機器人作為 AI 課程中專題。

就上述運用 Lego Mindstorms 在教學的方式，Lego Mindstorms 提供了一個具體的模擬情境，讓學生在設計、組裝、操作的過程中，了解教師所要傳授的觀念。

貳、 著重程式設計的模式

另一種取向是以介紹程式設計的觀念為主體 (Fagin, Merkle & Eggers, 2001; Barnes, 2002; Fagin & Merkle, 2003)，即教導學生了解程式設計中的主要概念，如循序控制、變數、常數等觀念，比較著重在操控機器人的程序，Lego Mindstorms 是用來讓學生了解程式的運作方式，讓學生從機器車的運作去了解程式運作的內在模式。使用機器人的模式來教導有關程式設計相關概念，並不是一個新穎的觀念，因電腦操控模式 (Computer Controlled Model) (Barnes, 2002) 早在 1980 年代，Logo 的 Turtle 和 Karel the Robot 就是使用這樣模式來教導程式設計的相關

原理，其中 Karel the Robot 隨著時間演進，所使用的程式語言也有各種的版本，從最早的 Pascal，再則是 C、C++，現在也有 Java 版本，但這些都是電腦螢幕上的虛擬機器人非實體機器人。因此使用實體的機器人除了因是控制一個實體機器人而使學生覺得有趣之外，它也提供學生實際觀察的執行中程式的管道 (Barnes, 2002)。

此外學者 (Lawhead, Duncan, Bland, Goldweber, Schep, Barnes & Hollingsworth, 2003) 依據 Stein (1998) 所提出資訊科學教學的方式，應從 "Computation is calculation" 模式轉換 " Computation is interaction" 的模式，在這種模式下，在一個程式中輸入不應該只是一個數值而應是一些可以被觀察到的事物，而輸出也不應是一個計算出來的數值應是一個不斷隨時間改變而改變的動作或狀態，因此他們認為機器人用於資訊科學的教學中是一個很好的選擇。針對不同的程式設計觀念發展出相對應的機器人操作情境，例如為了教導有關函數、參數、回傳值等觀念，機器人的操作情境是撰寫一個函數使得機器人可以左或右轉某個角度。

上述運用 Lego 的模式，就是運用 Lego Mindstorms 來建造一個抽象表徵機器模型，讓學生能透過這個表徵機器模式了解程式內在運作機制，正如 du Boulay (1989) 指出憑藉著一個有效的表徵機器使程式的運作對學生而言不是一個「黑盒子」 (black box) 而是一個「透明盒子」 (glass box)。而這也是 Mayer (1989) 證實一個有表徵機器的學生在解決某些問題上會比沒有的學生來得好。

Winslow (1996) 指出一個好的教學的策略是，在開始教學時教師應盡量將相關知識、規則、模式簡單化，然後當學生學會之後，再逐漸修改、擴充這些相關知識。此外在學習程式語言的相關語法後，如何整合使用這些語法和它們所蘊含的程式設計原則是程式教學中應予以特別注意的 (Linn & Dalbey, 1989)。Schep

及 McMulty(2002)認為運用 Lego Mindstorms 教導程式語言，可分為幾個步驟，首先是給學生一個短而完整的範例程式，讓學生了解程式的內容，並且能解釋這程式將會使機器人會有怎樣的動作。接著再給學生一個尚未完成的程式，學生需要依據題目的要求將程式完成，並確保機器人的運作是符合題意的。最後才讓學生根據一個特定情境設計一個完整的機器人操控程式，並且這程式能讓機器人能完成題目所要求的工作。