# 教育部教學實踐研究計畫成果報告

教育部教學實踐研究計畫成果報告
Project Report for MOE Teaching Practice Research Program

計畫編號/Project Number：PGE1090537
學門專案分類/Division：通識(含體育)
執行期間/Funding Period：109 學年度

## 計畫名：以問題導向學習視覺化程式來提升運算思維之研究

搭配課程：大一的院必修課程「程式設計與邏輯分析」(2 學分)

計畫主持人(Principal Investigator)：辛靜宜

執行機構及系所(Institution/Department/Program)：明新科技大學多媒體與遊戲發展系

成果報告公開日期：

□立即公開 ■延後公開(統一於 2023 年 9 月 30 日公開)

繳交報告日期(Report Submission Date)：

# 以問題導向式學習視覺化程式來提升運算思維之研究


## 摘要

本計畫於人文設計學院大一的「程式設計與邏輯分析」二學分課程，以視覺化程式問題導向教學幫助學生學會視覺化程式 Scratch，提升運算思維，進而接軌文本式 Python 程式語言。本計畫結果如下。

1. Scratch 問題導向教學可幫助學生學會 Scratch，提升運算思維。
2. 運算思維與程式（涵蓋 Scratch 與 Python）的學習成就相關性不大。研究者認為這是因為運算思維是程式的要件，包含理解程式與演算法等思考層面等，但程式的完成還有軟體與語法的執行層面。
3. Scratch 學習對於接軌 Python 學習的幫助取決於兩者運算思維是否相通，當相通（如迴圈），兩者學習成就呈中度相關，但因 Python 還有語法熟練等因素未達高度相關。若要以視覺化程式接軌文本式程式，兩者的運算思維要盡可能重合。
4. 不少學生在做 Scratch 專案時會自發的在藝術層面著墨，預期可將藝術融入程式設計來強化學習動機。

110 學年度本校強化資訊教育，將「程式設計與邏輯分析」的二學分課程改為三學分，主持人將因應時數增加，用研究結果提出改進方案。



關鍵字：問題導向學習、視覺化程式、運算思維

# Research on Problem-Based Learning on Visualized Programming to promote Computational Thinking

## Abstract

This project aims to help the freshman students of the college of Humanities and Design to learn Scratch, to improve students' computational thinking, and to adapt to the Python programming language in the two credit course “Programming Design and Logic Analysis”. The results are as follows.

1. Problem-based learning in Scratch can help students learn Scratch and improve their computational thinking.
2. There is little correlation between computational thinking and programming (including Scratch and Python) learning achievement. The researcher thinks that this is because computational thinking is essential to programming design, which includes concepts such as understanding of programming and algorithms, but the completion of programming also requires the execution of software and syntax.
3. Whether Scratch learning is helpful to integrate Python learning depends on how much computational thinking of the two programming languages coincident. When they coincident (such as loops), the two learning achievements are moderately correlated. They are not highly correlated because Python requires syntax proficiency. To adapt text-based programming by visualized programming, the computational thinking of the two should overlap as much as possible
4. Many students spontaneously focus on esthetic when doing Scratch projects. It is expected that blending art into programming designs can strengthen their learning motivation.

In the 110th academic year, due to the importance of information education, the school increased the 2 credits course "Programming Design and Logic Analysis" to 3 credits. In response to the increase teaching hours, the researcher will use the results of this project to propose improvements.

Key Words: problem-based learning, visualized programming, computational thinking

# 以問題導向式學習視覺化程式來提升運算思維之研究

## 一、 研究動機與目的(Research Motive and Purpose)

資訊素養是大學教育不可或缺的一環，也是不分科系所有大學生必須具備的。經濟合作暨發展組織（OECD）從 1997 年就進行大規模「關鍵素養的定義與選擇」（DeSeCo）的跨國研究中，定義出二十一世紀公民的三大關鍵素養，其中第一項就是互動式使用工具的素養，包含使用語言、符號和文本等知識的能力，也包含運用資訊和科技的能力。DeSeCo 並定義素養為：「能成功地回應個人或社會要求的能力」。因此素養不是知識，不是技能，而是包含個人獲取和應用知識、認知與技能的能力。換言之，大學應培育學生資訊素養，使學生能與時俱進，成功運用資訊，達成個人或社會的要求。

根據林育慈與吳正己（2016）的研究，目前我國資訊教育自 2009 年已進入運算思維（computational thinking）導向，強調利用電腦知識來培養學生邏輯思維與問題解決能力，培養學生資料處理技能的階段。強化運算思維，就能有效利用資訊科技解決問題，並可應用資訊科技到其他領域，因此運算思維導向的資訊教育不單是資訊相關科系所必備的能力，更是國民因應未來挑戰的核心素養。

然而主持人在教學現場，發現學生在文本式程式語言的學習中，不但無法寫出程式，甚至看不懂程式，而這源自於運算思維不足，因此於 109 學年度，進行本計畫，打算在人文設計學院的大一「程式設計與邏輯分析」的二學分課程，先以視覺化程式 Scratch 教學提升學生運算思維的能力，之後再接續文本式程式 Python 的學習。

## 二、 文獻探討(Literature Review)

### (一)、運算思維

運算思維可以簡單定義為：利用電腦解決問題時所必要的知識、態度與技能，是一個思考歷程，包含形成問題、組織解法，並將解法表示成運算工具能理解的形式（Wing, 2011）。根據教育部運算思維推動計畫網站（http://compthinking.csie.ntnu.edu.tw），運算思維是利用電腦科學的基本概念進行問題解決、系統設計與人類行為理解的思維模式（李恩萱、李忠謀，2018）。教育部運算思維推動計畫針對問題解決，訂出運算思維四向度是：

1. 拆解：將複雜的問題拆解成容易理解與分類的部分；
2. 樣式辨識：找出問題之間的相似之處；
3. 抽象化：將重要的部分列出，忽略不重要的部分；
4. 演算法：找到問題的解決步驟。

發展運算思維最有效的方法是學習電腦科學（K-12 computer science framework, 2016），程式設計則是展現運算思維的媒介（Grover & Pea, 2013），透過程式設計可實作運算思維的各種向度（林育慈、吳正己，2016）。教育部運算思維推動計畫網站也認為程式設計融入教學是培養運算思維的有效方法。

然諸多研究顯示：程式設計入門課若採用文本式的程式語言，往往導致教師與學生花費大量精力在程式語言特性（如語法結構），而非發展解題的設計方法，而目前取向是改用視覺化程式（李恩萱、李忠謀，2018）。呂永鈞（2015）就分別以文本式 Python 與視覺化 Blockly 教國小五年級學生，結果發現 Blockly 大幅降低程式語法的障礙，讓學生專注程式邏輯的學習，因而提升了運算思維的能力。Chao（2016）也認為視覺化程式因提供初學者視覺化物件的支持，有助於初學者理解程式架構，進而將心力集中在發展解題策略。

目前也有很多學校提出視覺化程式的運算思維導向課程，如在國中資訊科技概論將音樂

與運算思維導入程式教育（陳怡芬、林育慈、翁禎苑，2018），又如臺灣師範大學也開設通識課程「運算思維與程式設計」（李恩萱、李忠謀，2018）。在諸多視覺化程式中，Scratch 堪稱是目前最受歡迎的視覺化程式（劉長諺，2018），本研究也採用 Scratch 來提升運算思維。

然視覺化程式教學是否真有助於提升學生運算思維？要探討此問題，必須先定義如何評量運算思維。目前常用的有工具是國際運算思維挑戰賽與程式相關成績。

國際運算思維挑戰賽（Bebras）自 2004 年起舉辦，每年 11 月中在全球同步舉行，參賽者設定是國小三年級至高中三年級的學生，其宗旨是激發學生對資訊科學的興趣，並了解學生的運算思維能力。Bebras 題目分易、中、難三等級，全部都是情境題。我國自 2012 年起加入 Bebras，目前由國立臺灣師範大學資訊工程學系主辦（bebras.csie.ntnu.edu.tw）。

除 Bebras 之外，因程式設計則是展現運算思維的媒介（Grover & Pea, 2013），使用程式相關成績來做為評量運算思維能力的工具是很自然的選擇，除一般與課程相關的程式成績，目前也有程式分析網站可免費評分上傳的專案如 Dr. Scratch 與 Chippy 等自動程式評量系統。但自動評分系統並不核對原先題目，若學生上傳一個程式邏輯正確但不合老師題意的專案，仍有可能拿高分。

現回到視覺化程式教學是否真有助於提升學生運算思維？主持人進一步將問題細分為二並分別說明如下。

1. 視覺化程式教學是否真有助於提升學生的運算思維？目前提升結果並不明顯，會因受試者程度與教學法而異。
   (1) 當受試者年齡較大運算思維夠高時，提升空間就不大；但受試者年紀較小運算思維較弱時可望提升。（李恩萱、李忠謀，2018）針對國立臺灣師範大學大學生（n=238）就因 79%以上的學生在前測時答對率已 80 %以上，後續研究甚至不做後測，改用視覺化程式檢驗學生的運算思維能力。呂永鈞（2015）針對國小五年級學生的研究則發現視覺化程式教學可提升運算思維。
   (2) 講述教學對提升學生運算思維無幫助，但範例自學有幫助。劉長諺（2018）針對國立高中二年級學生，分別以講述教學（控制組，n=23）與範例自學（實驗組，n=23）教導 Scratch，其中運算思維分數是將測驗題 10 題依運算思維四個向度（拆解、樣式辨識、抽象化與演算法）分類，每一題橫跨多種向度而得四組向度分數，結果發現控制組的四個向度在前後測均無差異，實驗組在拆解、抽象化與演算法三個向度雖然也沒有差異，但在涵蓋樣式辨識的 10 題總分（即測驗總分）前後測達顯著差異。換言之，範例自學組在運算思維測驗總分前後測達顯著差異。
2. 運算思維測驗分數與 Scratch 成績是否正相關？目前研究結果並不明顯，主持人認為分歧的主因是評分方式。劉長諺（2018）的研究使用 Dr. Scratch 自動評分系統來打學生 Scratch 專案成績，運算思維測驗分數則如上段所言，是將學生測驗分數分成四個向度分數，此四個向度分數與 Scratch 專案分數並無相關性。劉長諺認為原因出在 Dr. Scratch 自動評分系統評分標準是以程式功能性為依歸，運用到越多樣性的程式積木得分越高，因此學生越熟練 Scratch 得分越高。

綜合以上文獻探討，雖然由學理上認為視覺化程式教學有助於運算思維的提升，且運算思維與程式學習相關，但目前並沒有量化研究的強力背書，這是因為易受教學法、評分方式與受試群體先備知識的干擾，因此研究者擬再次檢驗。本計畫受試群體為私立科技大學學生，不同於李恩萱與李忠謀（2018）針對國立大學生族群，預計沒有前測過高的問題。其次，本計畫不會使用講述教學，將用問題導向教學（於下下段說明）。再者，針對評分方式也與劉長諺（2018）不同：Scratch 專案成績則由教師評分，不用自動評分系統；運算思維分數則以直接以測驗分數計算，不細分向度，考題選擇以運用運算思維多種向度為主。

## (二)、視覺化程式 Scratch

視覺化程式也稱為積木程式，主要用圖形化的積木來取代複雜的程式語法，使用者只需填入相關的文字即可執行自己想要的程式。目前視覺化程式很多，如 Scratch、Blockly、APP Inventor 2 都是，也各具特色。

Scratch 是 MIT Media Lab 在 2006 年公開的積木程式，有網頁版與離線版，都是免費軟體，界面如下圖 1。目前在職場並不會直接以 Scratch 為開發工具，而是作為入門程式的資訊教育素材，堪稱各級學校中最人氣的視覺化程式（劉長諺，2018）。


圖 1　Scratch 界面圖

## (三)、問題導向式學習

根據國家教育研究院圖書館學與資訊科學大辭典（terms.naer.edu.tw/detail/1678753/），問題導向學習（Problem-Based Learning, PBL）於 1960 年代起源於加拿大的醫學教育改革，是一種以學習者為中心的課程設計與教學模式，老師設定教學目標後，透過真實的問題來引發學習者探討，老師只是從旁引導，並不會直接給定答案，學習者經由討論、思考而解決問題。傳統教學大多屬範例示範教學，與 PBL 之比較如下。

表 1　範例示範教學與問題導向式學習比較（陳毓凱、洪振方，2007）

| 教學法 | 教師與學生的角色 | 適用課程 | 特色 |
|---|---|---|---|
| 範例示範教學 | 老師示範，學生跟著演練，老師是課程主導者。 | 注重知識傳授的課程。 | 最具效率，學生對課程整體架構較有感。但低成就學生很可能只是死背知識，不會活用。 |
| 問題導向式學習 | 老師提出問題與指引，學生必須自己找解答，老師只是指引者或者學習促進者。 | 注重問題解決的課程。 | 學生比較能活用知識來解決問題。但知識傳授的效率較差，學生的知識架構也比較零碎或片段。 |

程式教學的終極目標就是培養學生透過程式設計解決問題（Winslow, 1996），因此程式設計與問題解決密不可分，教學設計上應該以問題解決為導向，再導入需要用到的相關程式

指令或語法（吳正己、林凱胤，1997），程式教學採用問題導向學習是自然的趨勢，而目前研究也顯示採用問題導向學習的程式教學成效不錯。陳佳宜與李忠謀（2018）針對明星高中女學生 272 人教視覺化程式 Code.org，其中實驗組（目標導向教學加問題導向教學）與控制組（傳統講述教學）各半，結果發現實驗組比對照組，在概念理解與程式實作表現都較好，組內的學習落差也較小。而陳明溥（2007）在文本式程式教學研究，以教學法（程序導向教學與問題導向教學）與教學內容（Quick Basic 與 Visual Basic）二因子實驗研究法，針對四組共 132 名高中生的研究顯示問題導向的 Visual Basic 教學有最佳的問題解決表現。

PBL 可依教學情境與教學目標調整，樣式也日趨多樣化，目前 PBL 常見的教學模式有以下種四種（楊坤原，2008）：

1. 醫學院模式：8-10 名學生的小班教學，以小組討論與個案分析為主。
2. 走動的促進者模式：全班分成若干組，每組 4-5 名學生，學生小組討論，導師則擔任走動的促進者，在小組間走動幫忙。
3. 同儕導師模式：類似走動的促進者模式，但多配置一些同儕導師擔任走動的促進者。
4. 班級模式教學：由教師主導 PBL 過程，但不要直接講述提供答案，而是經由提問指引學生自己找到答案。

本計畫教學內容是 Scratch，因 Scratch 只需很少的基礎知識就可以著手做專案，所以主持人採從做中學的模式。此外，因學生必須獨立完成程式且教學場域為一人一機的電腦教室，因此本計畫採班級的 PBL 教學。

## 三、 研究問題(Research Question)

本計畫旨在以 Scratch 問題導向教學增進學生運算思維，進而接續 Python 的學習，其具體研究問題如下。

1. 學生 Scratch 學習成就為何？
2. 本研究之教學可否提升學生運算思維能力？
3. 學生運算思維能力、Scratch 學習成就與 Python 學習成就三者是否相關？

## 四、 研究設計與方法(Research Methodology)

所有的研究問題皆以 Excel 2016 來處理，其統計方法如表 2。

表 2　本計畫資料處理與分析

| 研究問題 | 資料來源 | 統計分析 |
|---|---|---|
| 學生Scratch學習成就為何？ | Scratch 上機考、<br>Scratch 遊戲專題、<br>Scratch 作業 | 描述性統計<br>試題分析(鑑別度與難易度)<br>質性說明 |
| 本計畫之教學可否提升學生運算思維能力？ | 運算思維前後測分數 | 描述性統計<br>成對樣本t檢定<br>獨立樣本t檢定 |
| 學生運算思維能力、Scratch 學習成就與 Python 學習成就三者是否相關？ | 運算思維能力之前後測分數、<br>Scratch上機考成績、<br>Scratch作業總分、<br>Scratch專題成績、<br>下學期的Python上機考成績。 | Pearson相關分析 |

表 2 所列各項數據來源說一一說明如下。

## （一）、Scratch 上機考

Scratch 上機考是本課程期中考，題目取材自作業類似題。一題 25 分，總分 100 分，每一題僅有全對或有錯兩種計分方式，評分方式僅考慮積木邏輯，不考慮藝術層面。考題內容為條件判斷、變數、迴圈與流程控制，題目如下表。

表 3　Scratch 期中考上機考題目

| | |
|---|---|
| 1. | 貓咪沿著 X 軸來回走，X 軸上隨機出現球，貓咪遇到球喊 Go Go Go 去踢球讓球會飛上去，球遇到邊緣就反彈但落地就停止，球停止後貓咪喊 GG，程式結束。 |
| 2. | 每隔 0.5 秒上方隨機出現閃電往下掉，以左右鍵控制貓咪在 X 軸上走，貓咪被閃電電到就喊 GG，遊戲結束，遊戲分數就是生存秒數。 |
| 3. | 10 秒內，隨機出九九乘法的題目，答對就換新題，答錯就遊戲停止，以答對題數計分！ |
| 4. | 輸入 n，求 k， 其中 1*2+2*3+3*4+...+k*(k+1) >n　(詢問 n，說出 k) |

## （二）、Scratch 作業

本課程依進度出六次個人作業，題目是範例之類似題，一題五分共九題，由老師評分，除邏輯對錯正確與否，藝術層面也會加分。

## （三）、Scratch 遊戲專題

Scratch 遊戲專題是本課程期末評量，學生必須設計一款遊戲，於學期末報告，除邏輯對錯正確與否，也會考量整體規劃如畫面美觀、配樂與口頭報告。

## （四）、運算思維前後測

運算思維之前後測題目均取材 Bebras，題目本身已具備良好的鑑別度。計分方式也沿用 Bebras 計分方式，題目均為單選題。考量受試群體的背景，依高中一年級程度的難易度標準選題，第 1 題至第 7 題為低難度題目，第 8 題至第 12 題為中難度題目，第 13 題至第 15 題為高難度題目，總分是 280 分。運算思維之前後測分別於學期初與學期末施測。下表各列舉低（中、高）難度題目一題。

表 4　運算思維前後測低（中、難）難度題目各列舉一題

| | |
|---|---|
| 1. | 低難度。兩隻海貍想要啃倒三棵樹，每隻海貍啃倒一棵樹需要十分鐘。雖然海貍可以任意安排時間啃樹，但為了避免牙齒，兩隻海貍不能同時啃同一棵樹。假設可忽略海貍交換啃不同樹的間隔時間，請問兩隻海貍要啃倒三棵樹的最短時間為多久？　A）30 分鐘　B）20 分鐘　C）15 分鐘　D）10 分鐘 |
| 2. | 中難度。有一間飯店，裡面所有的房間號碼都是由左到右的兩個數字組成；第一個數字代表房間所在樓層，第二個數字代表從電梯到房間的步行距離。有個顧客上門想要住宿，但他不太喜歡走路，所以步行距離愈短的房間愈好。若有多間房間的步行距離相同，這位顧客比較喜歡樓層低一點的房間。請將下列飯店空房間的號碼，依照這位顧客的喜好排列：12，25，11，43，22，15，18，31，44，52。越左邊的號碼，對應這位顧客越喜歡的房間。請問下列何者的順序正確？　A）18，15，12，11，25，22，31，44，43，52　B）52，43，44，31，22，25，11，12，15，18　C）11，31，12，22，52，43，44，15，25，18　D）11，12，15，18，22，25，31，43，44，52 |
| 3. | 高難度。小海貍有一台很特別的電腦，它提供兩種指令來計算數學式子，這兩種指令的使用說明如下。R 指令：當 f 是一個數學運算加減乘除+−*／之一；而 x1, x2, ……, xn 代表一串數字，<br>那麼（Rf（x1, x2, ……, xn））會計算 x1 f x2 f……f xn，舉例來說，(R+(1, 2, 3, 4))將會計算 1+2+3+4，故結果為 10。M 指令：當 f 是一個函數，x1, x2, ……, xn 代表一串數字，那麼（Mf（x1, x2, ……, xn））會分別將數字代入函數而得數列 f(x1), f(x2), ……, f(xn)。舉例來說，當 q(x) = -x，那麼 (Mq(1, 2, 3, 4))將會 |

把數列 (1, 2, 3, 4)分別代入 q(x)，得到結果為數列(-1, -2, -3, -4)。現在假設 t(x) = 3x+2 且 q(x) = -x，請問下列式子
(R+(R+(Mt(0,2,4)))(R+(Mq(Mt(3,5))))會得到什麼結果？　A）7　B）0　C）-7　D）-4

## （五）、Python 上機考

Python 是本計畫結束之後（109-2 學期）的接續課程，當時期末考因疫情改為作業，所以本研究以期中上機考做為 Python 學習成就分數。上機考共五題，一題 20 分，總分 100 分，以五組測資點去測試程式是否正確。每一題有全對或有錯兩種計分方式。內容涵蓋基本輸入輸出、數學運算、條件判斷、迴圈、流程控制、函式與基本字串，題目如下表 6。下半學期教的串列、字串、集合、字典與檔案等則沒有涵蓋。

表 5　Python 上機考題目

| | |
|---|---|
| 1. | 某人的體重是 x 公斤，身高是 y 公分，輸入，輸入 x、y 數值，輸出 x 公斤相當於幾磅，y 公分相當於幾英尺幾英吋，以及某人的 BMI 指數。<br>一磅是 0.454 公斤，一英吋是 2.54 公分，一英呎是 12 英吋，BMI 指數是體重(公斤)除以身高(公尺)的平方。 |
| 2. | 請使用迴圈敘述撰寫一程式，讓使用者輸入一個正整數 n 與正整數 m，其中 n > m，利用迴圈計算並輸出排列數 P(n, m)的值。<br>P(n, m) = n!/ (n-m)!　或是 n*(n-1)* (n-2)*…..*(n-m+1)　共 m 個數相乘 |
| 3. | 請撰寫一程式，讓使用者輸入一個正整數 x，求 x 所有正因數的總和。<br>如輸入 12，輸出 28 (因 1+2+3+4+6+12 = 28)。 |
| 4. | 請撰寫一程式，讓使用者輸入一個正整數，將此正整數以反轉的順序輸出，但必須以常見的數字形式呈現。如輸入 305，則輸出 503；輸入 70800，則輸出 807。 |
| 5. | 請撰寫一程式，讓使用者輸入一個正整數 x，將 x 所有質因數從小到大一一列出。<br>如輸入 12，輸出 2, 3。 |

# 五、 教學暨研究成果(Teaching and Research Outcomes)

## （一）、教學過程

本計畫受試群體是 109 學年度明新科技大學多媒體與遊戲發展系日間部一年學生，原應為 45 人，但後因中途休學或資料不全(如問卷填答不齊等因素)，受試群體為 37 人。受試群體這 37 名學生約 2/3 來自設計學群，1/6 來自商管學群，1/6 來自電資學群。另外為避免前後測的練習效應，主持人特別情商本校資訊工程系大三選修應用統計學的學生自願來參與運算思維前後測來作為對照，人數為 20 人。這 20 名資訊工程系大三學生僅參與運算思維前後測，沒有修本門課程，也沒參與本計畫其他部分。本計畫採班級式的問題導向式教學，課程簡介如表 6。

表 6　本課程簡介

| | |
|---|---|
| 開課系(所) | 明新科技大學多媒體與遊戲發展系 |
| 中文課程名稱 | 程式設計與邏輯分析(2 學分) |
| 英文課程名稱 | Program Design and Logic Analysis |
| 課程屬性 | 系所必修(人文與設計學院) |
| 授課對象 | 大學部一年級學生 |
| 教學目標 | 以視覺化程式幫助同學增進運算思維，補強原本不足的資訊科技核心素養，並得以接續後面文本式程式(Python)課程。 |
| 教學方法 | 問題導向式學習 |

| 成績考核方式 | 回家作業、上機考、專題 | | | |
|---|---|---|---|---|
| | 週次 | 課程主題 | 內容說明 | 備註 |
| | 1 | 課程簡介 | | 運算思維能力前測 |
| | 2 | Scratch 簡介 | | |
| | 3 | 動作、外觀與造型 | 回家作業 1 | |
| | 4 | 「如果」與偵測類積木 Part 1 | | |
| | 5 | 「如果」與偵測類積木 Part 2 | 回家作業 2 | |
| | 6 | 問答、基礎運算與計時器 Part 1 | 回家作業 3 | |
| | 7 | 問答、基礎運算與計時器 Part 2 | 回家作業 4 | |
| | 8 | 運算與變數 | 回家作業 5 | |
| | 9 | 廣播、音效與分身 | 回家作業 6 | |
| | 10 | 調整進度 | | |
| | 11 | 期中上機考 | | |
| | 12 | 期中考檢討 | | |
| | 13 | 吃果子遊戲 | | |
| | 14 | 打磚塊遊戲 | | |
| | 15 | Scratch 遊戲設計企劃 | | 運算思維能力後測 |
| | 16 | Scratch 遊戲設計製作 | | Scratch 學習動機問卷調查(主持人自編) |
| | 17 | Scratch 遊戲專題報告 | | |
| | 18 | Scratch 遊戲專題報告 | | 教學反映問卷調查(本校統一調查) |

現舉一例如下說明本計畫班級式的問題導向式教學模式。教學內容是「如果」與偵測類積木。老師並不主動提出正確解答，而是經由問題指引與討論讓學生自己找出解決方法。問題：在 X 軸上隨機放一個蘋果，貓在 X 軸上隨機位置走來走去，若遇到蘋果就吃掉，提供一例如下表 7。

表 7　班級式的問題導向式學習之一例

| **老師提示** | **學生任務** |
|---|---|
| 需要幾個角色？ | 學生在 Scratch 介面設定舞台與兩個角色：貓與蘋果。 |

| | |
|---|---|
| 利用上次教的內容，讓貓走動來回兩次以便吃蘋果。 | 學生在 Scratch 介面寫貓走路的程式，如下圖。<br>  |
| 教師引領學生看偵測類積木與控制類積木。 | |
| 寫蘋果的程式，以便製造貓遇到蘋果，蘋果就消失（被吃掉）的效果。教師要學生試試看程式是否達到預期的結果。若不行的話，試著使用其他控制類積木，並想一下為什麼不行。 |  <br>學生在 Scratch 介面操作，常見的錯誤如下圖。<br>助教或老師指引以下正確方式，如下圖。最後要討論總結：建立學生程式是一連串的執行步驟，不是規則，學會程式的運算邏輯。<br>  |
| 教師請學生重複測試。 | 助教或老師指引必須加上外觀類積木，才會造成蘋果執行一次之後就不再出現，如下圖。 |


| 總結 | 結論與討論 |
|---|---|

## （二）、教學與研究成果

依表 2 進行統計分析，首先將相關數據之敘述統計列表如下。

表 8　本計畫各項分數的敘述統計摘要表（n＝37）

| | 平均 | 標準差 | 最大值 | 最小值 | 總分 |
|---|---|---|---|---|---|
| 運算思維前測 | 134.35 | 43.70 | 209 | 49 | 280 |
| 運算思維後測 | 166.11 | 38.70 | 265 | 65 | 280 |
| Scratch上機考 | 68.92 | 27.26 | 100 | 25 | 100 |
| Scratch遊戲專題 | 73.65 | 10.45 | 95 | 60 | 100 |
| 六次作業總分 | 27.32 | 7.66 | 43 | 13 | 45 |
| Python上機考 | 67.57 | 14.42 | 100 | 40 | 100 |

### 1. Scratch 學習成就

針對研究問題 1，由 Scratch 三項成績來看，學習成就良好，大部分學生能運用 Scratch 基本技能並活用，學生甚至自發性花時間在 Scratch 作業與遊戲專題的專案的藝術層面，研究者認為這是由藝術引發的學習動機使然。

在總分 100 的上機考，學生平均 68.92，考量每一題只區分全對或錯，無部份給分，大部分學生已具備 Scratch 基本技能。學生的答題型態如表 9，每一題的難易度與鑑別度如表 10。其中難易度即受試群體 37 人的答對率，鑑別度則是高分組的答對率減低分組的答對率。因整體受試群體只 37 人，且分數僅有 25、50、75 與 100 四種，所以高低分組依平均分數劃分，即高於平均 68.92 的是高分組（75 分與 100 分）有 22 人，低於平均 68.92 的是低分組（25 分與 50 分）有 15 人。

表 9　Scratch 上機考的答題型態

| 答題型態 | 分數 | 人數 |
|---|---|---|
| 對 1 | 25 | 6 |
| 對 1, 2 | 50 | 6 |
| 對 1, 3 | 50 | 3 |
| 對 1, 2, 3 | 75 | 10 |
| 對 1, 2, 3, 4 | 100 | 12 |
| 總人數 | | 37 |
| 平均 | | 68.92 |
| 標準差 | | 27.26 |

表 10　Scratch 上機考的難易度與鑑別度

| | 第一題 | 第二題 | 第三題 | 第四題 |
|---|---|---|---|---|
| 難易度 | 1.00 | .76 | .68 | .32 |
| 鑑別度 | .00 | .32 | .80 | .55 |

表 9 最特別的是答題集中到五種型態。這是因為程式學習具累積性，此外本考題是以作業為範本，因此題目越貼近作業，屬越前面的題材，學生答對率越高。由表 9 也可看出，可看出第一題最簡單，其次是第二與第三題，第四題最難，此與表 10 所列難易度吻合。

在總分 45 的 Scratch 作業，平均 27.32，標準差 7.66；在總分 100 的 Scratch 遊戲專案，平均 73.65，標準差 10.45，表現不錯。學生遊戲專案相當多樣化，有改編課堂教學的打磚塊遊戲、接果子遊戲與問答遊戲，也有取材自己常玩的小遊戲如迷宮、變裝遊戲。比較特別的是，很多學生在做作業或專題時，不只注重積木的編排，也在藝術層面投入大量心力，這應該是學生由設計的興趣引發的學習動機。如遊戲專案「小車車長大」（https://youtu.be/fYUQsCxn2eM），遊戲本身其實很簡單，但畫面安排的很溫馨可愛。又如作業「火箭升空」（https://youtu.be/Pssc1VHgS7Q），只利用簡單的積木堆疊就製造出炫麗的動畫效果，全靠學生慢慢去調整配樂與火箭運動的協調性。也有學生展現出整體的遊戲企劃，先有故事鋪陳再進入遊戲，整個作品「月球掃蕩計畫」版（https://youtu.be/3qOgQKZ_lTQ），如真實遊戲開發之縮小版。

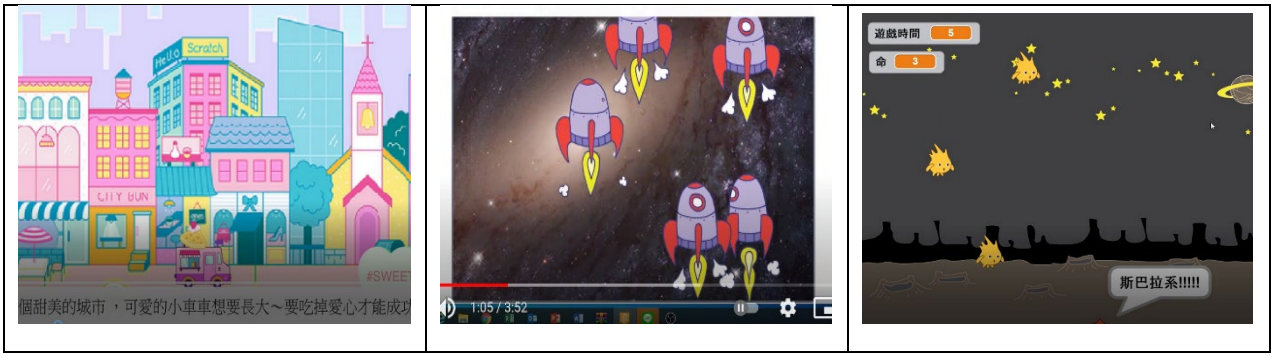


圖 2　學生 Scratch 專案截圖：「小車車長大」、「火箭升空」、「月球掃蕩計畫」

## 2. 運算思維前後測

針對研究問題 2，本計畫受試群體的運算思維有所提升。為避免練習效應，主持人特別由本校資訊工程系大三學生找尋自願者，參與運算思維前後測來作為對照組，人數為 20 人。這 20 名資訊工程系大三學生僅參與運算思維前後測，沒有修本門課程，也沒參與本研

究其他部分。

如表 11，針對同一組的前後測，研究者以成對樣本 t 檢定檢驗，結果發現受試群體 37 人，後測平均較前測平均高 31.76 分且達顯著差異；但在對照組 20 人，前後測未達顯著差異。此外，研究者分別針對前測與後測，以獨立樣本 t 檢定，檢驗受試群體 37 人與對照組 20 人是否有組別差異，結果顯現受試群體 37 人與對照組 20 人於前測有組別差異，但在後測差無組別差異，如表 12。

表 11　運算思維能力前後測成對樣本 t 檢定統計表

| | 運算思維能力 | 平均 | 標準差 | 最大值 | 最小值 | 自由度 | t 值 | p |
|---|---|---|---|---|---|---|---|---|
| 受試群體 37 人 | 前測 | 134.35 | 43.70 | 209 | 49 | 36 | | |
| | 後測 | 166.11 | 38.70 | 265 | 65 | 36 | | |
| | 後測-前測 | 31.76 | 48.11 | 166 | -54 | 36 | 4.02 | < .001 |
| 對照組 20 人 | 前測 | 160.75 | 47.07 | 280 | 49 | 19 | | |
| | 後測 | 171.50 | 48.56 | 280 | 40 | 19 | | |
| | 後測-前測 | 10.75 | 34.98 | 56 | -59 | 19 | 1.374 | .093 |

表 12　運算思維能力前後測獨立樣本 t 檢定統計表

| 測驗 | 組別 | 人數 | 平均 | 標準差 | p |
|---|---|---|---|---|---|
| 運算思維能力前測 | 受試群體 | 37 | 134.35 | 43.70 | |
| | 對照組 | 20 | 160.75 | 47.07 | .045 |
| 運算思維能力後測 | 受試群體 | 37 | 166.11 | 38.70 | |
| | 對照組 | 20 | 171.50 | 48.56 | .671 |

結合以上二表，主持人認為對照組一開始運算思維比受試群體好（前測有組別差異），但後來受試群體慢慢追上對照組（平均後測比前測進步 31.76 分且達顯著差異），而對照組雖有練習效應而進步 10.75 分但未達顯著差異，最後兩組在後測也無組別差異。如圖 3 所示。

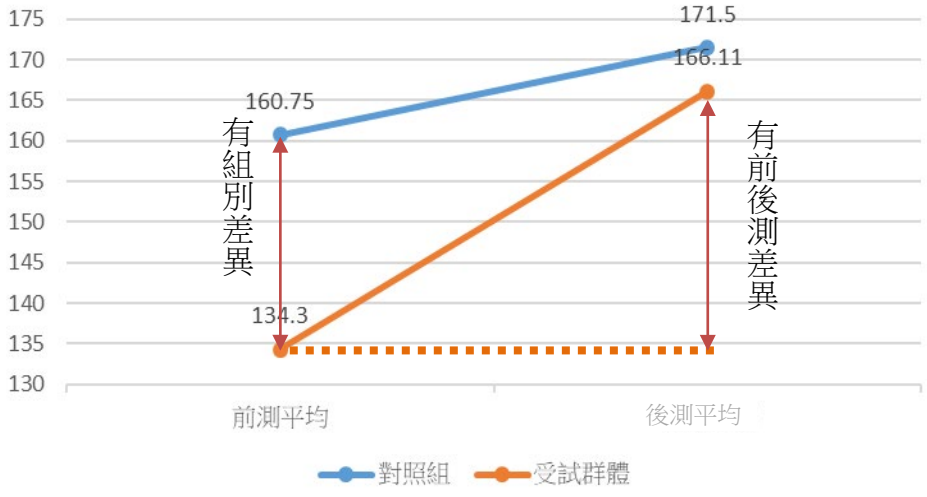


圖 3　運算思維前後測

以對照組而言，以上結果與李恩萱與李忠謀（2018）針對國立大學大學生的研究算是吻合，運算思維達到一定程度之後就不大容易提升，因此前後測未達顯著差異。

以受試群體而言，以上結果與劉長諺（2018）針對國立高中二年級學生的研究部分算是吻合。劉長彥以範例自學教學，的確可以幫助學生於運算思維前後測分數達統計上差異。

研究者認為當受試群體運算思維不是特別高時，視覺化程式搭配合適的教學法（如範例自學、PBL，但非傳統講述教學）可提升學生運算思維。

### 3. 運算思維能力、Scratch 學習成就與 Python 學習成就三者的相關性

針對研究問題 3，依表 2 計算相關係數如下，本計畫各項分數的相關性並不高。

表 13　本研究計畫各項分數之相關性

| | 後測 | Scratch 上機考 | Scratch專題 | Scratch作業 | Python上機考 |
|---|---|---|---|---|---|
| 前測 | .32 | .26 | .37 | .27 | .15 |
| 後測 | | .00 | .40 | .29 | .10 |
| Scratch 上機考 | | | .13 | .29 | .40 |
| Scratch 專題 | | | | .53 | .48 |
| Scratch 作業 | | | | | .47 |

由上表討論如下。

(1) 由表 13 觀之運算思維前後測與 Scratch 各項成績、Python 上機考僅低相關。除運算思維後測與 Scratch 專題相關係數達 0.4 之外，其他的相關係數都小於 0.4。此點與劉長諺（2018）研究吻合，雖然劉長諺的計分方式與本研究略有出入，但仍出現大同小異的結果。研究結果出爐前，主持人原本認為若改變評分方式有可能出現運算思維與程式成績相關的結果，但顯然必須修正。如劉長諺所言，學生越熟練 Scratch，其程式分數越高，因此程式分數顯現的不只式思考層面，還有執行層面。運算思維是程式的要件，包含理解程式與演算法等思考層面等，但完成程式還有對軟體與語法的執行層面，因此運算思維分數與程式相關成績相關性偏低。

(2) Scratch 各項成績與 Python 上機考中度相關。主持人認為這代表 Scratch 對接軌 Python 有些幫助。本次 Scratch 教學內容與 Python 上機考大部分重合，運算思維相同，都涵蓋條件判斷、變數、迴圈與流程控制，因此學生在 Python 時對這些觀念的程式邏輯相對熟悉，但 Python 還必須學會對語法的理解與熟悉，所以對學生有些幫助但兩者沒有高度相關。另一方面，Python 上機考平均 67.57（表 8），表面上學生應該也具備 Python 基本技能，但這只是期中考。在 Python 下學半學期，教學內容為串列、字典與檔案等等，是學生在 Scratch 沒有接觸過的概念，研究者明顯看到學生學的很吃力，甚至放棄學習。因此研究者認為，視覺化程式學習對文本式程式學習的關鍵在於兩者的運算思維是否相通，當相通時，兩者達中等相關，但因涉及語法熟練所以沒到高度相關。現以一例：輸入一正整數 n，求 n 所有正因數總和（即 Python 上機考考題 3，答對率 96%），說明運算思維、Scratch 與 Python 三者之間的關聯性如下圖，可看出三者思路相同，但 Scratch 要加上對積木的熟悉才能找到合適的積木並堆疊出正確的順序，而 Python 要加上對語法的熟練才能選用合適的指令。

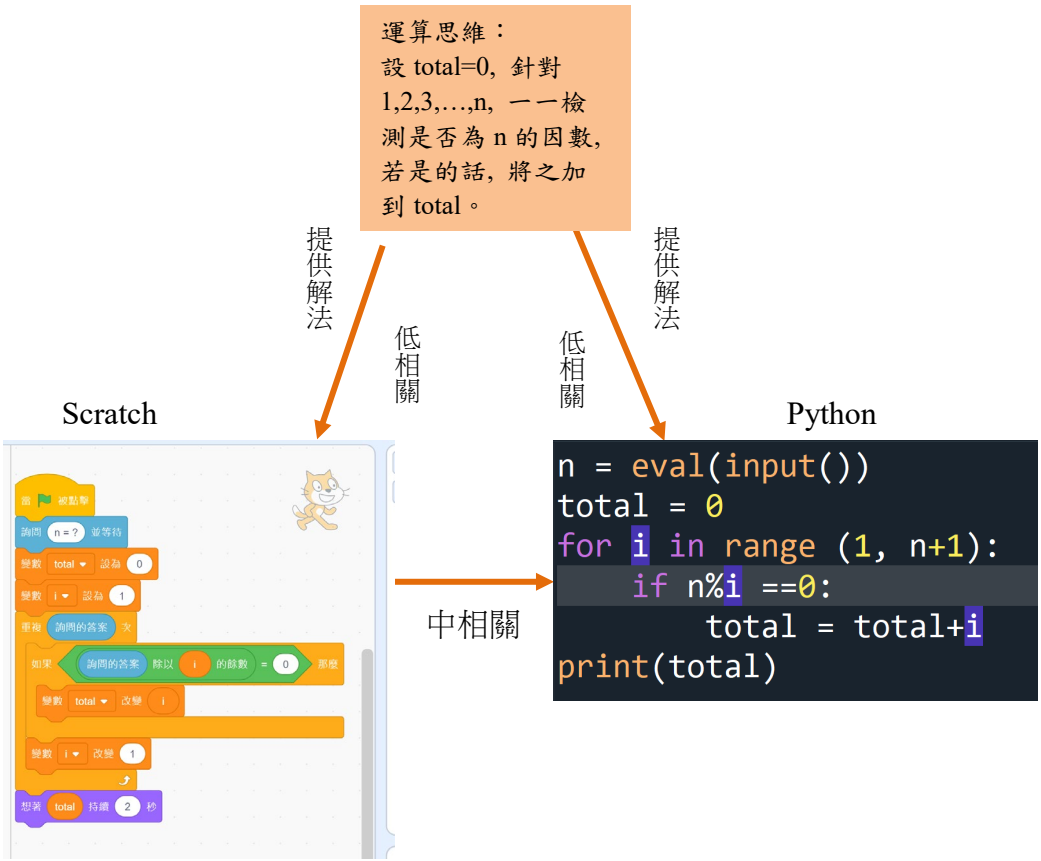


圖 4　一例說明運算思維、Scratch 與 Python 三者之間的關聯性

(3) Scratch 上機考、Scratch 專題與 Scratch 作業三者之間倆倆的相關係數，除專題與作業之外，都只低度相關。雖然 Scratch 上機考題目幾乎都是由作業改編，但作業與專題若在畫面編排的美觀上多所著墨，老師會酌予加分，而上機考純以程式邏輯計分。

## （三）、教師教學反思

私立科技大學人文設計學院大一學生問題導向式的 Scratch 教學設計，可幫助學生學會 Scratch，增進運算思維，學生甚至額外在程式中加入設計的藝術層面，但有兩點必須注意。

1. 運算思維與程式（涵蓋 Scratch 與 Python）的學習成就相關性不大。研究者認為這是因為運算思維是程式的要件，包含理解程式與演算法等思考層面等，但完成程式還有對軟體與語法的執行層面。
2. Scratch 學習對於接軌 Python 學習的幫助取決於兩者運算思維是否相通，當相通（如迴圈），兩者學習成就呈中度相關，但因 Python 還有語法熟練等因素未達高度相關。然視覺化與文本式程式運算思維越貼近應越有幫助。

## （四）、學生學習回饋

固然本計畫教學有所成效，但學生觀感如何呢?主持人參考 Pintrich, Smith, & Mckeachie (1989)根據 Bandura 的社會認知的學習理論編制的「學習動機與學習策略量表」(Motivated Strategies for Learning Questionnaire，以下簡為 MSLQ)，自編六點評分(1: 非常不符合，2:不

符合，3:稍不符合，4:勉強符合，5: 符合，6: 非常符合)的問卷，於學期做調查。MSLQ 量表不但有良好的信度與效度，在美國也常作為診斷學習之工具（吳靜吉、程炳林，1992）。調查結果摘要如以下兩表。

表 14　學習 Scratch 的問卷調查統計表

| 向度 | 題目(1-6 計分) | 平均 | 標準差 | α 信度 |
|---|---|---|---|---|
| 學科價值 | 我覺得 Scratch 幫助我建立程式的觀念。 | 4.73 | 0.80 | |
| | 我覺得使用 Scratch 的經驗有助於我學習其他軟體或程式語言(如 Python, C sharp, Unity)。 | 4.51 | 0.90 | |
| | 我覺得 Scratch 很實用。 | 4.41 | 1.07 | |
| | 我認為 Scratch 很有趣。 | 4.59 | 0.96 | |
| | 學科價值 | 4.56 | 0.74 | 0.80 |
| 積木的掌控力 | 作 Scratch 專案時，我花很多時間想程式架構或堆疊積木的邏輯。 | 4.68 | 1.00 | |
| | 作 Scratch 專案時，都會預先想好角色的動作再來堆疊積木。 | 4.46 | 0.73 | |
| | 作 Scratch 專案時，角色通常都會按照我心中的設定跑動。 | 4.32 | 1.06 | |
| | 積木的掌控力 | 4.48 | 0.74 | 0.70 |
| 自我效能 | 我有信心做出 Scratch 的專案。 | 4.43 | 0.93 | |
| | 我做出 Scratch 的專案時很有成就感。 | 4.57 | 1.04 | |
| | 我願意花時間做 Scratch 作業。 | 4.38 | 0.95 | |
| | 自我效能 | 4.46 | 0.83 | 0.80 |
| | 問卷 | 4.50 | 0.67 | 0.91 |

表 15　學生對學習 Scratch 的心得(開放式問題)

| 編號 | 學生評語 |
|---|---|
| 1 | 剛開始學 Scratch 還算簡單，可是到後面就越來越複雜，期中考甚至不及格。後來期末有一項作業是要自己做一個遊戲然後介紹，做的途中他一直不聽使喚，死都不會讓分數往上跑，就在某天他終於會動了，當下真的很開心，比中樂透還開心，也很有成就感，真滴讚! |
| 2 | 做完自己想呈現的狀態很有成就感，老師很有耐心><! |
| 3 | Scratch 讓我建立程式的觀念，當我自己做出遊戲時，真的很有成就感。 |
| 4 | 有時候做出的成品跟同學們分享可以得到樂趣與成就感，能夠訓練自己的程式概念，我覺得很讚! |
| 5 | 學習 Scratch 有助我熟悉程式語言的基本觀念。使用 Scratch 時以積木代替程式區塊以及能及時得到結果的成就感有助提升我學習程式語言時的興趣與信心。 |
| 6 | 我覺得練 Scratch 很好玩，不但能訓練邏輯，當作成功時還能很有成就感 |
| 7 | 可以讓我們學習到程式的基礎概念，以及執行的順序 |
| 8 | 我覺得上 Scratch 可以讓我們用簡單的方式去了解程式設計。 |
| 9 | Scratch 我很喜歡只是上課上的太少，想多一點這種課程。 |
| 10 | 我覺得 Scratch 很有趣也很好玩，我之後也願意再繼續學 Scratch! |
| 11 | 有時候會卡住，但老師聰明美麗又有耐心，讓我學得很順利!也讓我對程式有基本的了解!!><THX! |
| 12 | 我覺得 Scratch 很好玩，可以再多嘗試看看 |
| 13 | 我覺得學這程式很好玩，有非常多的程式讓自己去運用 |
| 14 | 能學習到這個程式軟體，能讓我學習到很多知識。 |
| 15 | 老師上課認真，上課內容生動有趣 |
| 16 | 簡單使用，角色可愛，滿好用的 |
| 17 | 我覺得 Scratch 非常有趣 |
| 18 | 滿有趣的 |
| 19 | 好玩! |
| 20 | 還蠻好玩的 |
| 21 | 謝謝老師的教導! |
| 22 | Scratch 至少看得懂，很好。 |

| | |
|---|---|
| 23 | 動腦一點都有趣 |
| 24 | 因為 Scratch 可以做動畫，寫起來比較不無聊。 |
| 25 | 雖然對我來說沒有很簡單，但努力學起來後，覺得很開心 |
| 26 | 學到很多技巧，但不熟練所以不太會用 |
| 27 | Scratch 比那種要自己寫的程式簡單多了 |
| 28 | 他的反彈很神奇，偶爾會卡在上面。 |
| 29 | 我覺得 Scratch 看似簡單，但是操作起來對我來說很困難! |
| 30 | 初嘗 Scratch 稍有不懂，但還是可以學習，雖不怎麼有趣但是值得去學。 |
| 31 | 真的好難….我只會畫貓貓 |
| 32 | 那討厭的臭貓，害我差點被當!! |
| 33 | 好難喔 QQ |

由表 14 觀之，本問卷信度 0.71—0.91，算是可以接受的問卷，大體上學生給予本課程稍微正面但不是非常正面的評價，但大致上認為有助於學習程式，且簡單好玩，做出想要的效果時有成就感，現說明如下。就量化數據而言，學生在學科價值、積木的掌控力與自我效能各項平均落在 4.32—4.73，本問卷為六點計分，因此各項得分約落在勉強同意與同意之間，算偏正面並不是非常正面的回應。其中平均得分最高的是「我覺得 Scratch 幫助我建立程式的觀念。平均為 4.73」，而平均得分最低的是「作 Scratch 專案時，角色通常都會按照我心中的設定跑動。平均為 4.32」。主持人由教學現場與學生的相處與對學生的觀察，認為學生普遍自信心低，可能導致問卷得分偏低。

在問卷中平均得分最高的「我覺得 Scratch 幫助我建立程式的觀念。平均為 4.73」也在表 15 的開放式心得中得到佐證，如綠底所示，有五位同學主動提及 Scratch 幫助程式的學習。

此外在表 15 的開放式心得中，有六名同學提到成就感或信心(黃底) ，有十名同學提到樂趣或有趣或好玩(藍底) ，主持人認為成就感應該是一部分來自對積木的掌控力，此與運算思維以及後續的程式學習相關，另外一部分成就感來自對自己的滿意，而這可能與藝術層面相關。以下列本課程之學校例行的教學反應，總平均 4.41，算是教學成效良好的課程。

表 16　本課程教學反映問卷之選擇題部分

| 題號 | 題目 | 平均值 |
|---|---|---|
| 1 | 開學時，教師曾就本科目之教學目標、授課大綱、教材、評分方式等作說明。 | 4.55 |
| 2 | 教師上課很少遲到、早退或中途離席。 | 4.43 |
| 3 | 教師未曾缺課或有事未能到課時，會事前禿之學生並安排事後補課。 | 4.43 |
| 4 | 教師上課的態度認真，具教學熱忱。 | 4.43 |
| 5 | 本課程教材與內容有組織且強大，課程內容與目標相符。 | 4.38 |
| 6 | 教師上課時，在教科書外，會適當補充講義或其他教學資料。 | 4.28 |
| 7 | 作業能配合教學內容，且份量與難易適中，教師有批閱作業。 | 4.45 |
| 8 | 教師上課時，表達清晰、條理分明、容易理解。 | 4.35 |
| 9 | 教師上課時，能鼓勵學生發問討論，師生互動良好，並營造引導學習之環境。 | 4.43 |
| 10 | 教師能考慮到學生之需求並做調整 | 4.38 |
| 11 | 教師對考核及評分公正合理。 | 4.48 |
| 12 | 教師對學生之評量能反映教學重點，測出學習效果。 | 4.35 |
| 13 | 本課程幫助學生知能之成長，提升學生在此領域之能力。 | 4.35 |
| 14 | 整體而言，本課程之教學品質優良。 | 4.45 |
| | 課程平均 | 4.41 |

表 17　本課程教學反映問卷之開放式文字填填答部分

| |
|---|
| 學得很多 |
| 沒有意見 |
| 有 BUG 真的會爆氣 |

| |
|---|
| 老師水喔<3 |
| 對我來說有點難 |
| 很棒！超喜歡勿 |

## 六、 建議與省思(Recommendations and Reflections)

110 學年度本校加強資訊教育，將「程式設計與邏輯分析」的二學分課程改為三學分，主持人由本研究結果，利用多的時數，提出改進方案。

1. 教學內容改為 App Inventor 2。App Inventor 2 也是視覺化程式，可利用此軟體做出 Android 可使用的 App，但觀念比 Scratch 複雜，也較貼近 Python 的觀念。既然教學時數增加，應可改用 App Inventor 2 取代 Scratch 作為教學內容。Scratch 與 App Inventor 2 的比較如下表。

表 18　Scratch 與 App Inventor 2 之比較

| | Scratch | App Inventor 2 | 評比 |
|---|---|---|---|
| 屬性 | 視覺化程式 | 視覺化程式 | 平手 |
| 實用性或應用範圍 | 適合做動畫與遊戲 | 做 App | App Inventor 2 勝 |
| 使用軟體 | 免費的網路版軟體 | 免費的網路版軟體 | 平手 |
| 跨域性 | 畫面編排可加上藝術素養 | 畫面編排可加上藝術素養 | 平手 |
| 分享或展示性 | 使用者也必須有 Scratch 軟體 | 只要用 Android 手機，掃描 QR code 即可使用，不須下載 App Inventor 2 軟體 | App Inventor 2 勝 |
| 學習內容 | 大部分專案只須學會基礎的條件判斷、迴圈，再加上計時器與廣播的運用就可以 | 除基礎的條件判斷、迴圈，還要加上檔案與網頁資料擷取，另外也可搭配手機的三軸加速度感測器等 | App Inventor 2 學習內容較多 |

2. 教學法引進改用 STEAM 跨域課程設計。除原先的 PBL，也將藝術導入課程，為 App 畫面編排增色，希望藉此保有學生的學習動機。

## 參考文獻(References)


吳正己、林凱胤（1997）。問題解決導向的程式語言教學。*資訊教育雜誌創刊十年特刊*，75-83。

吳靜吉、程柄林（1992）。激勵的學習策略量表之修訂。*測驗年刊*，**39**，59-78。

李恩萱、李忠謀（2018）。教育學院大學生運算思維與程式設計學習成就。在中華民國教育部主編，*TANET 2018 臺灣網際網路研討會論文集*（頁 2592-2596）。桃園市：國立中央大學。doi:10.6861/TANET.2018.0480

呂永鈞（2015）。*藉由國小五年級學生學習程式設計探究運算思維能力在 Bebras 測驗上的表現*（未出版碩士論文）。國立臺灣大學，臺北市。

林育慈、吳正己（2016）。運算思維與中小學資訊科技課程。*國家教育研究院教育脈動電子期刊*，**6**，1-13。

陳怡芬、林育慈、翁禎苑（2018）。運算思維導向程式設計教學–以「動手玩音樂」模組化程式設計為例。*中等教育*，**69（2）**，127-141。

陳明溥（2007）。程式語言課程之教學模式與學習工具對初學者學習成效與學習態度之影響。*師大學報：科學教育類*，**52**，1-21。

陳佳宜、李忠謀（2018）。從目標導向到問題導向：高學習成就學生之程式設計學習研究。在中華民國教育部主編，*TANET 2018 臺灣網際網路研討會論文集*（頁 2586-2591）。

桃園市：國立中央大學。doi:10.6861/TANET.2018.0479
陳毓凱、洪振方（2007）。兩種探究取向教學模式之分析與比較。*科學教育月刊*，**305**，4-19。
楊坤原（2008）。問題本位學習的理論基礎。*問題導向專題式教學研討會論文集*，**1**，5-9。
劉長諺（2018）。*視覺化程式學習對於高中生提升運算思維能力之影響*（未出版碩士論文）。國立交通大學，新竹市。
Chao, P. (2016). Exploring students computational practice, design and performance of problem-solving through a visual programming environment. *Computer & Education*, **95**, 202-215.
Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Research*, **42(1)**, 38-43.
Pintrich, P. R., Smith, D. A. F., & Mckeachie, W. J. (1989). A Manual for the use of The Motivated Strategies for Learning Questionnaire (MSLQ). Ann Arbor, Michigian: National Center for Research to Improve Teaching and Learning, School of Education, The university Michigan.
K-12 Computer Science Framework Steering Committee. (2016). K-12 computer science framework.
Wing, J. M. (2011). Research Notebook: Computational Thinking—What and Why? The Link. *The magazine of the Carnegie Mellon University School of Computer Science.*
Winslow, L. E. (1996). Programming pedagogy: A psychological overview. *SIGCSE Bulletin,* **28**, 17-22.