

即時適性化架構於運算思維課程設計之應用

黃雅惠

摘要

隨著教育部於 108 年課綱將「科技領域」納入為第八大領域的必修課，在科技領域中的資訊科技旨在培養學生具有創造思考、批判思考、問題解決、邏輯與運算思維(computational thinking)等高層次思考的能力。希望學習者能將自己所學的知識與能力運用在動手實作及跨學科的結合，如科學、科技、工程、數學 (Science, Technology, Engineering, and Mathematics, STEM)。同時在 108 年課綱也強調著，教育應依據學習者的差異給予不同的學習路徑，也就是孔子的教育理念「因材施教」的作為，因此教育需要教育者與學習者進行互動。本研究將使用貝氏網路 (Bayesian network) 來符合適性化(Adaptive)架構於運算思維課程設計的應用。研究對象為未學習過程式教育或邏輯概念的國小中高年級學生，經由設計過的邏輯概念的關卡，依據學習者的學習狀況給予最適合他們的關卡題目，來讓學習者了解基礎的運算思維概念並加以結合各概念來達到學習目的，並減少學習者的學習盲點。

關鍵字:運算思維、STEM、適性化、貝氏網路

Application of Instant Adaptation Architecture in the Design of Computational Thinking Courses

Ya-Hui, Huang

Abstract

With the Ministry of Education's 108 years of curriculum, required course includes “ science and technology field ” in the eighth field. Information technology in science and technology field aims to cultivate students with higher-order thinking abilities, such as creative thinking, critical thinking, problem solving, logic and computational thinking. It is hoped that learners can apply their knowledge and abilities to hands-on and interdisciplinary integration, such as Science, Technology, Engineering, and Mathematics (STEM). At the same time, the 108-year curriculum emphasizes on giving different learning paths to learners based on individual differences, which comes from Confucius's educational philosophy about teaching students in accordance with their aptitude. Therefore, education requires educators to interact with learners. In this study, we use the Bayesian network to apply the adaptive architecture in the design of computational thinking courses. The participants in this research is elementary and middle-level students who have not studied procedural education or logical concepts. Using the level of the designed logical concept, researcher can give the most suitable level to learners. It not only helps them to understand the basic computational thinking, combine basic concepts to achieve learning goals also reduce their blind spots in learning.

Keywords : computational thinking, STEM, adaptation, Bayesian network

一、緒論

教育部於 108 年課綱將「科技領域」納入為第八大領域的必修課，而科技領域包含了生活科技與資訊科技兩門課程。課程旨在培養學生的科技素養，透過運用科技工具、材料、資源，進而培養學生動手實作，以及設計與創造科技工具及資訊系統的知能，同時也涵育創造思考、批判思考、問題解決、邏輯與運算思維等高層次思考的能力。放眼國際，諸多先進國家亦設有科技領域，強調科學、科技、工程、數學及設計等學科知識的整合運用，藉由強化學科間知識的連結性，來協助學生理解科學與工程的關連。因此透過科技領域的設立，將科技與工程之內涵納入科技領域之課程規劃，藉以強化學生的動手實作及跨學科，如科學、科技、工程、數學（Science, Technology, Engineering, and Mathematics, STEM）等知識整合運用的能力。（十二年國民基本教育課程綱要-國民中小學暨普通型高級中等學校,105）。科技領域中的資訊科技則以運算思維為主軸，透過電腦科學相關知能的學習，培養邏輯思考、系統化思考等運算思維，並藉由資訊科技之設計與實作，增進運算思維的應用能力、解決問題能力、團隊合作以及創新思考的能力。國中的資訊科技課程也在這十二年國民基本教育課綱下，成為了必修課。資訊程式因其學習的困難度相較於其他科目高並且我們應該著重於「邏輯與運算思維」而非程式語言上。所以為了降低其學習難度，我們將使用 Code.org 作為其主要運算思維課程設計的訓練關卡，用以降低學生學習新課程的排斥力，也可減少學習程式語言繁瑣的規矩與錯誤率。

二、文獻探討

本研究目的在探討適性化(Adaptive)架構於運算思維(Operational Thinking)課程設計之應用，而適性化架構是運用貝氏網路(Bayesian Network)給予學習者適合之學習路徑，學習者在解關卡的過程中可以發現到自己的學習盲點(Learning Blind Spots)，並針對概念上的錯誤給予修正。學習者的學習會有自我的學習能力、興趣、人格特質或認知風格等的不同，而以上因素會導致學習者的學習成效會有所差異。因此，本章為探討本專題參考的相關研究，整理出以下的文獻資料，作為本研究的依據。

(一) 適性化

適性化(Adaptive)是依據學習者的狀態或個人特質，來調整學習內容或呈現方式(Gagne 2005)，來符合學習者的個人化需求，也就是孔子的教育理念「因材施教」。然而每一位學習者，都有其個別差異，像是學習能力、興趣、人格特質或認知風格的不同，對於學習的成效也會有所差別。所以更應該以學習者的個別差

異作為設計的考量(廖怡雯 2014)。而適性化學習的特色有(1)教材取代部分教師對學習者所提供的教學活動。(2)根據學生的反應主動建議下一步。(3)支持不同的學習方式，讓學習者不受到限制。(4)讓學習者自主學習，不會與其他同學做比較，並提升學習速度(張皓程 2017)。

許多研究顯示，在符合學習者個別學習能力的環境下進行學習，可以有效地增進其學習成效(Dochy 1992; Moerkerke 1996; Weibelzahl 2001)。如圖 1 由教育科技應用的觀點，因每個人的大腦思維方式不同，所以學習能力、興趣、人格特質或認知風格皆都不一樣，因此將適性化學習的模式分為三大類：(1)個人化的學習路徑。(2)個人化的學習內容。(3)個人化的呈現方式(黃國禎,蘇俊銘&陳年興 2012)。

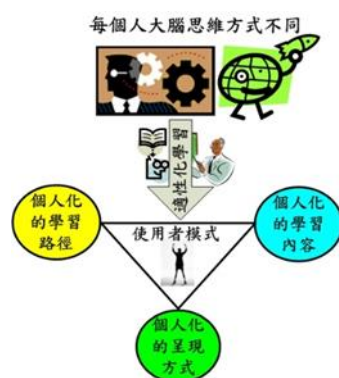


圖 1 適性化學習模式

適性化除了指個人化適性的學習內容之外，系統的規劃還是要符合可適性化的回饋內容。也就是教學教材應提供有意義的回饋給學習者，這些適當的回饋內容，不僅可以鼓勵學生持續往後的學習，也可以導引學生正確的學習方向以學習正確的概念。Elawar & Corno (1985)提出當學生答對或做出正確回應時，系統提供獎勵的回饋可改善學習者在該課程的學習成就與態度，也更加進一步說明適性化回饋機制具有以下的特性：(1)回應確認內容(Confirmation feedback)，意即簡單回應學習者的答案是否正確。(2)提供正確答案的回饋，也就是除了第一種簡單回應學習者是否正確外，還會進一步提供正確答案是什麼。(3)具有解說的回應，意即一步一步的解釋或者告訴學習者為什麼不符合答案或概念模糊的原因來提供錯誤相關回饋(Bug-related Feedback)。即經由分析學習者解題的結果及程序，依照學習者錯誤程度或出錯的原因，給予引導或提示，以協助正確地解決問題與修正迷思概念。(4)為了避免學習者過度依賴數位教學系統給的回饋資訊，以及避免學習者缺乏問題解決或摘要的能力，因此個人化適性的學習內容與回饋機制，不建議直接提供正確答案，應該要考量到學習者的程度。例如：對於高能力的學生，不需要提供明顯的提示；而對於低成就的學生，提供有步驟引導的提示(黃國禎,蘇俊銘&陳年興 2012)。

適性化可以應用的範圍很廣，而以下的學者們對於適性化做出了許多研究：林

淑文(2010)建立了以「試題反應理論」與「遊戲式學習」為基礎之適性化筆順測驗系統，用來提升學習者的學習興趣，也更有效地提升學習者「國字筆順」的能力。王昌斌與吳俊毅等人(2003)，則利用知識管理、電子化學習與代理人理論，結合貝式網路理論，建立了「個人化主動推播理論」，主動推播潛在的知識給學習者，使學習平台的功能更具多元與實用性。陳俊德等人(2006)提出以知識地圖與貝式定理相結合的適性化學習系統，利用知識地圖中的概念與權重進行分析，分析學習者在學習當中所缺乏的概念，同時配合學習者所自訂的學習目標和系統所儲存的紀錄，配合貝式定理的分析、機率預測，挑選出最適合學生的教材，此機制可以改善傳統迷思概念的學習診斷方法有效地進行回饋，並結合過去的學生經驗推薦，來提升學生的學習成效。黃國豪等人(2007)提出結合 Keefe & Monk 等人的學習風格量表(Learning Style Profile)與所羅門學習風格，Keefe & Monk 等人的學習風格量表的目的是在於判斷學習教材難易度與測驗難易度；所羅門學習風格則是用來分析教材類型與學習模式，希望透過適當的風格、教材及困難度的調整，來提升學生的學習成效。Wang & Huang(2008)提出利用改良過的螞蟻演算法，建立學生的適性化學習系統，不同的學習路徑會產生出不同的學習類型(learning style)，此系統著重在不同的學習類型所產生的課程教材之間的關聯性，每位學生的最佳化學習路徑皆不相同，這些不同的課程教材之間可能存在著很高的關聯性。Chen & Lee(2005)等人提出 Item Response Theory 搭配最大概似估計法(MLE)，利用 MLE 來評估學生的能力，再利用 Item Response Theory 的方法找出適用的個案，加以設計出適合學生能力的課程教材，此外還使用問卷的方式進行程度的微調，提升學生的學習成效。Huang & Chen(2007)則是使用基因演算法(GA)來進行適性化學習，評估每個課程的難易度與教材之間的關聯性，再配合基因演算法的適應函數進行子代的挑選，選出最適合學生的課程教材順序安排，此外也搭配 Case-Base Reasoning(CBR)來進行調整，計算出最佳化的學習路徑。最後以資訊教育來說，對於程式語言，研究發現學生的學習風格偏好對學生程式語言學習成效有影響，例如不同認知風格學生在產出 Scratch 動畫或遊戲作品的構圖層面上，視覺導向的學生表現顯著比語文導向的學生好(張素芬，2010)。上述研究皆證實適性化的學習可以為學習者帶來更有效地學習。

(二) 貝氏網路

貝氏網路(Bayesian Network)，又稱為信賴網路(Belief Network)、因果網路(Causal Network)、推論網路(Influence Network)、因果機率網路(Causal Probabilistic Network)、機率推論圖(Probabilistic Influence Diagram)和知識地圖(Knowledge Map)(Charniak 1991; Pearl 1986; Xenos 2004)。貝氏網路是一種有方向性的圖形模式(Directed Graphical Models)，但它不允許循環存在(Directed Acyclic Graph, DAG)，也就是有向非循環的圖形模式。舉例來說，貝氏網路可以用來表示學習結果(此班學生學習狀況好或不好)和其相關學生分數的機率關係；倘若已知某種狀況下，貝氏網路就可以被用來計算及預估各種可能學生分數之發生機

率。

貝氏網路是以貝氏定理(Bayes' Theorem)為基礎發展而來，以圖形表示網路結構，用條件機率(Conditional Probability)的方式來獲得新的資訊去修正事前機率，而得到事後機率的方法，用來描述變數之間的關係。使用貝氏網路後，可以在不確定的條件下進行推理，結合直覺跟經驗上的優勢，用數學的模式呈現，並且可以用一致的角度，判斷初始機率影響下的不確定性推論。剛開始，貝氏網路應用於遺傳基因學理論，後來應用在許多領域中，像是醫療診斷(Antal et al. 2000)、故障診斷(Kirsch & Kroschel 1994)、食品的設計(Corney 2000)、衛星測試設備(Przytula & Thompson 2000)、導航器系統、人工智慧、決策支援系統、認知科學、電腦科學、工程界及柴油引擎火車頭等領域，應用的層面非常廣泛。在教育方面，Xenos(2004)曾用貝氏網路來預測及評價學生的行為。在遠距離教學的模式下，利用學生過去的經驗，建構大學電腦課程的學生行為的貝氏網路。利用這個貝氏網路，可以預測學生的未來行為，幫助教師在教育的過程中做決策，也能評估學生現行的狀況，讓教師及時發現學生的錯誤，或學習的盲點，適時的提出糾正，掌握學生的情況，以便將來能增進學生學習的效率，避免錯誤的一再循環。Arroyo & Woolf(2005)則使用教學系統上的日誌檔，分析學習者的行為，再利用貝氏網路來推論學生認知及情緒上的狀態，做為發展能自我改進學生的學習模式、教學，及不斷地能夠調適教學上的決策學習系統(謝錦泉 2007)。用貝式網路來診斷學習者的學習風格及知識學習的盲點，因為學習者的學習風格會隨著時間、環境等其他因素而改變。透過我們的系統，貝氏網路可以從學習歷程中即時反應出學習者的學習盲點，提供最符合學習者學習的教材，更能增加學習者的學習成效。

(三) 運算思維

運算思維(Operational Thinking)一詞最早是由 Carnegie Mellon University (CMU) 的電腦科學學者 Wing 所提出。她認為運算思維是普遍人們都可適用而非專屬於電腦科學家，且重要性等同於閱讀、寫作、算術等須具備的基礎能力。運算思維包含電腦科學基礎的概念，讓我們在生活解決問題時更有效率，也可在面對複雜困難的問題時，可透過抽象化、分解問題、模組化等方式，將問題重新規劃成我們熟悉的問題並加以解決(李威霖 2016)。Wing 在 2006 提出「運算思維是運用電腦科學基本概念來解決問題、設計系統、以及瞭解人類的行為，是每個孩童都應具備的基本素養，以及未來生活必備的電腦科學技能與知識」。之後又更進一步定義運算思維為「一種思考歷程，是規劃問題(formulating problem)與解決方案的心智活動，而這些問題解決方案能由人、電腦或兩者的結合來實施」(Wing 2011)。

運算思維的概念被 Wing 學者正式提出後，引發各國相關領域與組織熱烈的關注與探討，也讓許多學者認同運算思維的重要性並針對運算思維提出不同的見解。Bundy(2007)提出運算思維影響了所有領域的研究，同時也改變我們的思維模式，運算的概念提供了描述假設與理論的一種新語言，延伸了我們認知能力，想了解

21 世紀必須先了解運算思維。Lu 及 Fletcher(2010)指出運算思維是利用電腦科學基本概念來解決問題與設計系統的一種方法，使用不同層次的抽象與演算法概念，以更有效地理解問題與解決問題。Kazimoglu 等多位學者(2012)歸納各學者的論點，提出五點運算思維的核心技能：(1) 問題解決(Problem Solving):指的是定義問題、分解問題、評估適當的解決方案，最後發展抽象化。(2) 建立演算法(Building Algorithm):指有結構化且逐步執行流程解決特定問題，並且可多次使用於解決類似的問題。(3) 除錯(Debugging):為分析問題中的錯誤，是種批判性和程序性的思考。(4) 模擬(Simulation):也稱為建立模組(Building Models)，主要以事先設計或執行模組來幫助決策問題。(5) 社交(Socializing):指的是在上述問題解決、建立演算法、除錯和模擬等階段時，與人進行協調、合作或競爭的行為。

根據以上學者們對於運算思維的研究可以發現到他們注重於問題解決的能力，而對於運算思維的定義有些許的不同。所以美國 ISTE(International Society for Technology in Education)與 CSTA(Computer Science Teachers Association)合作，透過問卷調查的方式徵詢電腦科學相關的老師、學者和從業人員的回饋，而歸納出運算思維的操作型定義(ISTE 2011)。該研究指出運算思維是在問題解決的過程時包含以下的特性：(1) 把問題規劃成可用電腦或其他工具解決的形式。(2) 有邏輯地組織和分析資料。(3) 以抽象化表示資料。(4) 建立演算法將解決方案自動化。(5) 分析各種可能地最有效的解決方案。(6) 將問題解決過程標準化以解決其它的問題。另外，以上特性會受到學生個人特質不同而有所影響，例如：(1) 面對複雜事情的信心。(2) 面對困難問題的堅持度。(3) 容忍度。(4) 處理開放性問題的能力。(5) 與他人溝通合作達成共同目標的能力。

Google 於 2010 年建立的 Exploring Computational Thinking 網站也引用了上述 ISTA 與 CSTA 的合作研究，並致力於推廣運算思維，提供豐富的教案、影片和資源，幫助教師在現有課程融入運算思維教學。Google 對於運算思維的定義為「問題解決過程中所包含的一些特性(例如能有邏輯地排序與分析資料後，以詳列步驟的方式建立演算法來解決問題)及態度。例如能有信心去面對處理複雜且開放性的問題」。

Grover 和 Pea(2013)總結相關研究後，提出目前大家最廣為接受的運算思維包括:Abstractions and pattern generalizations (including models and simulations)、Systematic processing of information、Symbol systems and representations、Algorithmic notions of flow of control、Structured problem decomposition (modularizing)、Iterative, recursive, and parallel thinking、Conditional logic、Efficiency and performance constraints、Debugging and systematic error detection。(莊惠淇 2014)

綜合上述研究，運算思維普遍被認為是解決問題的過程，而這過程中牽涉到許多電腦科學相關基礎概念的運用，許多運算思維概念必備技能中，其中以「分解問題」被認定為最重要的技能。然而如何讓學習者學習運算思維是目前最困難的工作，傳統的教學大都以常見的程式語言(C、Java、Python.....)工具培養運算思

維的概念。讓初學者藉由學習程式語言的過程學到重要的概念，但此種方法對中小學學生而言過於艱澀難懂(Wing 2008)。因此在學習運算思維概念方面，Kelleher 及 Pausch(2005)建議以拖拉式的視覺化程式積木(Scratch、Microbit、Code.org……)較適合培養學生的運算思維概念，讓學生不必費心於拼字或語法上的錯誤，更能專注於程式設計中邏輯與結構上的思維培養。簡單來說，學習運算思維的工具應該是要具備低門檻、功能強大、鷹架支持、可遷移、及永續等特性(Repenning, Webb, & Ioannidou 2010) (李威霖 2016)。藉由圖形與視覺化頁面讓學生們較能輕易地了解運算思維，結合他們現有生活經驗上的知識，培養他們如何解決問題的能力。

(四) 學習成效

學習成效在這裡是指學生的學習程度，本專題採用 Carroll(1963)教授所提出的一種學習理論。他認為學生的學習成果是與其有效投入學習的時間成正比，它們彼此之間的關係，可以使用下列的數學函數來表示其概念「學習程度=學習實際所花的時間/學習真正所需的時間」(公式一)。亦即，學習程度等於學生實際花在學習上的有效時間，除以真正學好該學科所需要時間的函數。

而這學習程度以學生本身的特質和教學品質這二者最為重要。其中，「學習實際所花的時間」是由學生的「毅力」(Perseverance)，也就是學生願意且主動投入學習的時間量和「學習機會」(Learning Opportunity)，也就是學校提供學習的時間量，意即上課的時數兩者所決定；而「學習真正所需的時間」，則是由學生對該學科的「學習速度(即性向)」、「教學品質」和學生「瞭解學習的能力」三者所決定。因此，上述公式一可以再詳細表示為公式二(毛連塹&陳麗華 1991)(余民寧 2016)。

因此，Carroll 認為學習是沒有所謂「學習失敗」的學生，只有「學習快慢」的學生之分(余民寧 2016)。

$$\text{學習程度} = f\left(\frac{\text{學習實際所花的時間}}{\text{學習真正所需的時間}}\right) \text{ 公式一}$$

$$\text{學習程度} = f\left(\frac{\text{毅力} + \text{學習機會}}{(\text{學習速度}) + (\text{教學品質} + \text{了解學習的能力})}\right) \text{ 公式二}$$

(五) Code.org

Code.org 為一個非盈利組織，創建者為 Hadi Partovi 兄弟，他們也創建了與該組織同名的網站，其宗旨在於支持美國的學生學習計算機科學，也就是我們所知的資訊工程。該網站為教育機構，以促進和提高其學校的計算機科學課程為目的，提供免費編程經驗。在 2013 年 12 月 9 日至 2013 年 12 月 15 日，他們成功舉行一個全國性的「編程一小時 2013」(hour of code)宣傳週活動，來促進該國的計算機科學。

三、研究方法

(一) 研究架構

透過上述研究目的、研究範圍，我們將採取以下的方法、流程(如圖 2)。

1 研究工具

我們將使用 Code.org 的積木進行教學，為了降低程式語言學習的障礙，圖形化積木可以解決學習程式語言的限制與規則

2 研究對象

本研究的研究對象為無學習過程式語言或運算思維與邏輯概念的國小中高年級學生，為了證實適性化的架構對學習是有效的，實驗則會分為對照組與實驗組。

3 實驗流程

本研究使用「適性化架構」的方式於運算思維課程設計，目的在探討學生在運算思維概念的學習成效。若是學生能夠理解此堂課的概念，則進行下一個進階或是另一個新的概念；若是學生不能夠理解此堂課的概念，則在此觀念多加複習讓他到能夠理解，依據學生的學習狀況去做調整。

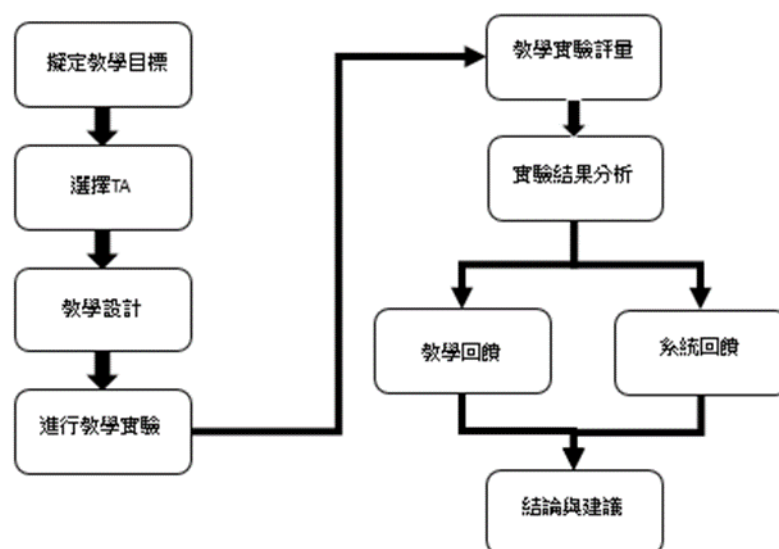


圖 2 實驗流程圖

4 分析結果

課程結束後，針對實驗組與對照組學生的回答狀況，了解「貝式網路」是否對學習有成效去做分析，取得最後的結論

(二) 遊戲介紹

現今有許多線上運算思維課程的教材，例如：美國麻省理工學院(MIT)開發的Scratch、Google 創建的Blockly games、以遊戲式學習聞名的CodeCombat 以及Hour of Code 的Code.org。Scratch、Blockly games、CodeCombat 與Code.org。這些線上教材都有共同的特色，為了讓初學者對於學習感到興趣且剛開始設計的題目難度不能太高，而他們的遊戲皆是從最簡單的次序概念，再到迴圈概念、條件概念……，最後整合以上的概念來達到遊戲的目標。如同學者 Kelleher 及 Pausch(2005)建議以拖拉式的視覺化程式積木培養學生的運算思維概念，讓學生不必費心於拼字或語法上的錯誤，更能專注於程式設計中邏輯與結構上的思維培養。簡單來說，學習運算思維的工具應該是要具備低門檻、功能強大、鷹架支持、可遷移、及永續等特性(Repenning, Webb, & Ioannidou 2010) (李威霖 2016)。

本研究的實驗遊戲為Code.org，Code.org 為一款使用 JavaScript 建置的闖關遊戲。各個課程可以針對不同年級或年齡給予不同的任務，而我選擇的是課程 3 中的迷宮部分關卡來進行教學與測驗。此課程網站是為了設計給 8-18 歲的學習者進行運算思維的學習，所以本研究對象為國小未學習過程式設計或運算思維概念的學習者。此遊戲的角色為一名殭屍，為了走到太陽花的過程中會遇到重重阻礙，像是食人花、牆壁、有限制的積木量……，所以要透過程式積木與概念的結合才能克服關卡、達到目標，就能解鎖下一道關卡。

(三) 實驗流程

本實驗流程分為三個部分(如圖 3)。第一部分為開始前會先進行志工訓練，因本研究需要人工方式紀錄學習者的分數、解題時間，與停留在此關卡的次數，在志工訓練過程中會告知志工本研究的目的、測驗規則與各關卡的注意事項。第二為在學習者部分，我們會進行基礎運算思維概念教學，學習者可以從教學中學到次序，迴圈，條件，以及各概念結合的複合題型。第三部分為進行實驗，與學習者解說完關卡規則與總測驗時間後，方能進行實驗測試，在實驗部分，分為有經過貝式網路的實驗組與未經貝式網路的對照組。



圖 3 實驗流程

1 第一部分:

志工訓練-實驗組

實驗組志工需了解志工手冊內容，並依照以下步驟進行實驗:

總共有十二道關卡，每一道關卡都會有關卡圖片，積木元件以及題目解答

實驗組志工執行步驟:

- 第一步:依據每一關卡，拿出關卡圖片，並依據該關積木元件拿出並給學生測試
- 第二步:將學生回答時間記錄下來(例如:1:05)
- 第三步:第九關卡以前測驗時間為兩分鐘，超時則重新開始計算此關卡，並記錄此關卡重複測驗幾次
- 第四步:第九關開始測驗時間為四分鐘，超時則重新計算此關卡，並記錄此關卡重複測驗幾次
- 第五步:若是學生回答超過時間，字卡不用更動；而到新關卡(不論是跨關還是倒退)，字卡需放回原位
- 第六步:詳細答案在此關卡投影片的下頁，依據學生的答案給予分數，並輸入成績到輸入器/PPT，他會告訴你下一道關卡

若是有任何問題，通知此負責人

志工訓練-對照組

對照組志工需了解志工手冊內容，並依照以下步驟進行實驗:

總共有十二道關卡，每一道關卡都會有關卡圖片，積木元件以及題目解答

對照組志工執行步驟:

- 前五個步驟和實驗組一樣
- 第六步:若是學生在此關卡超過十次，則在此關卡次數登記為 10，分數為 0，直接到下一關卡

2 第二部分:學習者教學


使用 Code.org 為程式設計課程教學工具，將學生分為實驗組與對照組，探討貝式網路對運算思維課程設計的應用。本學習者教學課程擬定為一個半小時，前四十分鐘為概念的介紹與操作，休息十分鐘後，剩下的四十分鐘為測驗時間。學生可以從教學中學到次序，迴圈，條件，以及各概念結合的複合題型。

本研究考慮到偏鄉教育的網路設備不普及，在教學與實驗上會遇到困難，所以使用實體積木教具進行實驗，解決設備不足的問題。

關卡教學


次序、for 迴圈關卡教學範例題目(如表 1)

表 1 關卡教學(次序、for 迴圈)

關卡教學題目	解法一	解法二	解法三
	<p>當運行時</p> <p>重複 2 次</p> <p>執行 移動-向前</p> <p>轉向-右方</p> <p>重複 2 次</p> <p>執行 移動-向前</p> <p>轉向-右方</p> <p>重複 2 次</p> <p>執行 移動-向前</p>	<p>當運行時</p> <p>重複 3 次</p> <p>執行 移動-向前</p> <p>移動-向前</p> <p>轉向-右方</p>	<p>當運行時</p> <p>重複 3 次</p> <p>執行 重複 2 次</p> <p>執行 移動-向前</p> <p>轉向-右方</p>


while 迴圈、條件關卡教學範例題目(如表 2)

表 2 關卡教學(while 迴圈、條件)

關卡教學題目	解法一	解法二	解法三
	<p>當運行時</p> <p>移動-向前</p> <p>移動-向前</p> <p>轉向-左方</p> <p>移動-向前</p> <p>移動-向前</p> <p>移動-向前</p> <p>移動-向前</p> <p>移動-向前</p>	<p>當運行時</p> <p>移動-向前</p> <p>移動-向前</p> <p>轉向-左方</p> <p>循環直到</p> <p>執行 移動-向前</p>	<p>當運行時</p> <p>循環直到</p> <p>執行 移動-向前</p> <p>如果左邊有路</p> <p>執行 轉向-左方</p>

條件中的條件(如表 3)

表 3 關卡教學(條件中的條件)

關卡教學題目	解法
	<p>當運行時</p> <p>循環直到</p> <p>執行 如果前面有路</p> <p>執行 移動-向前</p> <p>否則 如果右邊有路</p> <p>執行 轉向-右方</p> <p>否則 轉向-左方</p>

3 第三部分:進行實驗

以此關卡(如表 4 第九關)為例

表 4 第九關

<p>關卡設計目標: 基於各種不同狀況來做出決定</p>	
<p>題目鑑別度: 這道題目學生需辨別如果發生某個事件時, 就要做出與原本動作不同的決定</p>	
<p>關卡題目:</p> 	<p>本題積木元件:</p> <ul style="list-style-type: none">  x1  x9  x2  x2  x4  x2
<p>最佳解答:</p> 	

第四章 研究分析

此次實驗為臺東多間公立國小中高年級的學生，測驗者皆為未學習過程式設計或運算思維概念的學習者。實驗的評分標準本研究將套用 Carroll(1963)教授所提出的學習理論，他認為學生的學習成果是與其有效投入學習的時間成正比，它們彼此之間的關係，可以用下列的數學函數來表示其概念「學習程度=學習實際所花的時間/學習真正所需的時間」，所以本研究的實驗組與對照組都會套用此公式「 Σ 分數/解題時間*停留在此關卡的次數」就可以知道此學生的學習成效。實驗組會應用貝式網路來實現適性化的教學，藉由 IBM SPSS Modeler 工具，可以知道經貝式網路的實驗組學生適合的下道關卡題目。

如圖 4 為本研究的貝氏網路路徑圖，以第一關為例，若是學習者能在第一關得到分數為 5 分，就會到達第三關；若是 3 分，則到第二關；若是 0 分，則依然停留在第一關。依序為圖 4 表示

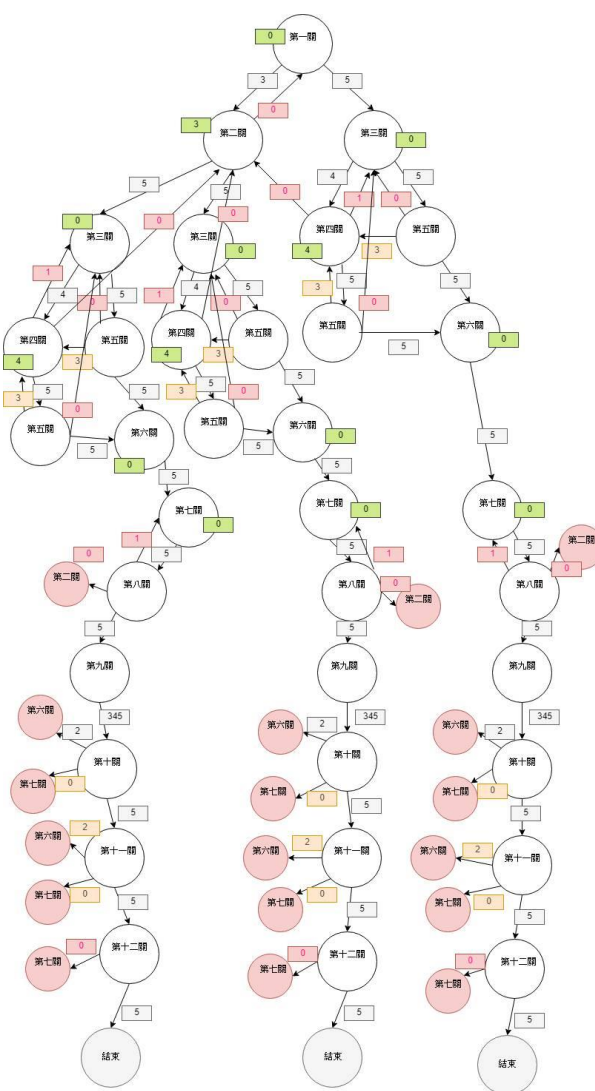


圖 4 貝氏模擬圖

依據以上的貝氏網路路徑圖，使用 IBM SPSS Modeler 貝式網路工具，得到以下的結果。因第九關為複合題型，會依照前面八道關卡的概念是否清楚就能知道能否在此道關卡輕鬆解題。

如表 5，第九關時間的條件機率，可以大致看出分數小於等於 451.7 時，時間皆在 46.2 秒內就完成作答，在此題停留的次數為 1~2 次；而分數大於 451.7 時，時間皆超過 46.2 秒才完成作答，在此題停留的次數為 1 次。由此可知，分數越高者，解題時間雖長，不過回答狀況較佳，相對地在第九關的停留次數也較少。

表 5 第九關時間的條件機率表

第九關時間 的條件機率						
上層		機率				
第九關次數	第九關分數	<= 26.6	26.6 ~ 46.2	46.2 ~ 65.8	65.8 ~ 85.4	> 85.4
<= 1.5	244.43	0.00	1.00	0.00	0.00	0.00
<= 1.5	311.65	0.00	0.00	0.00	1.00	0.00
<= 1.5	435.68	1.00	0.00	0.00	0.00	0.00
<= 1.5	439.18	0.00	1.00	0.00	0.00	0.00
<= 1.5	488.36	0.00	0.00	1.00	0.00	0.00
<= 1.5	497.25	0.00	0.00	0.00	1.00	0.00
<= 1.5	528.37	0.00	0.00	0.00	0.00	1.00
<= 1.5	940.43	0.00	0.00	1.00	0.00	0.00
<= 1.5	1108.07	0.00	0.00	1.00	0.00	0.00
> 1.5	451.7	1.00	0.00	0.00	0.00	0.00

如表 6，第十關時間的條件機率，可以大致看出分數小於等於 499.18 時，時間大部分在 44.8 秒內就完成作答，與第九關的關係為在第九關的停留的次數為 1~2 次；而分數大於 499.18 時，時間大致在 88 秒內完成作答，在第九題停留的次數為 1 次。由此可知，分數較高者，解題時間較短，在第九關的停留次數也較少。

表 6 第十關時間的條件機率表

第十關時間 的條件機率					
上層		機率			
第九關次數	第十關分數	<= 44.8	44.8 ~ 88	88 ~ 131.2	> 131.2
<= 1.5	254.77	0.00	0.00	1.00	0.00
<= 1.5	338.92	1.00	0.00	0.00	0.00
<= 1.5	473.18	1.00	0.00	0.00	0.00
<= 1.5	499.18	1.00	0.00	0.00	0.00
<= 1.5	512.36	0.00	1.00	0.00	0.00
<= 1.5	520.33	1.00	0.00	0.00	0.00
<= 1.5	535.87	0.00	0.00	0.00	1.00
<= 1.5	960.43	0.00	1.00	0.00	0.00
<= 1.5	1148.07	1.00	0.00	0.00	0.00
> 1.5	486.99	1.00	0.00	0.00	0.00

如表 7，第十一關時間的條件機率，可以大致看出分數小於等於 524.12 時，時間在 143.7 秒內就完成作答，與第九關的關係為在第九關的停留的次數為 1~2 次；而分數大於 524.12 時，時間均皆不同就完成作答，在第九題停留的次數為 1 次。

表 7 第十一關貝氏條件機率表

第十一關時間 的條件機率					
上層		機率			
第九關次數	第十一關分數	<= 58.2	58.2 ~ 92.4	92.4 ~ 143.7	> 143.7
<= 1.5	272.42	0.00	1.00	0.00	0.00
<= 1.5	373.21	1.00	0.00	0.00	0.00
<= 1.5	505.61	1.00	0.00	0.00	0.00
<= 1.5	524.12	0.00	0.00	1.00	0.00
<= 1.5	526.49	0.00	0.00	0.00	1.00
<= 1.5	542.54	0.00	0.00	0.00	1.00
<= 1.5	549.18	1.00	0.00	0.00	0.00
<= 1.5	975.43	0.00	1.00	0.00	0.00
<= 1.5	1178.07	1.00	0.00	0.00	0.00
> 1.5	497.61	0.00	0.00	1.00	0.00

如表 8，第十二關時間的條件機率，可以大致看出分數小於等於 535.06 時，時間均不一致才完成作答，與第九關的關係為在第九關停留的次數為 1~2 次；而分數大於 535.06 時，時間大都在 33.6 秒內就完成作答，在第九題停留的次數為 1 次。由此可知，分數較高者，解題時間較短，在第九關的停留次數也較少。

表 8 第十一關上層貝氏條件機率表

第十二關時間 的條件機率						
上層		機率				
第九關次數	第十二關	≤ 33.6	33.6 ~ 60.2	60.2 ~ 86.8	86.8 ~ 113.4	> 113.4
≤ 1.5	288.86	0.00	0.00	1.00	0.00	0.00
≤ 1.5	403.98	0.00	1.00	0.00	0.00	0.00
≤ 1.5	518.38	0.00	1.00	0.00	0.00	0.00
≤ 1.5	535.06	0.00	0.00	0.00	0.00	1.00
≤ 1.5	555.87	0.00	0.00	0.00	1.00	0.00
≤ 1.5	609.18	1.00	0.00	0.00	0.00	0.00
≤ 1.5	609.84	1.00	0.00	0.00	0.00	0.00
≤ 1.5	1018.28	1.00	0.00	0.00	0.00	0.00
≤ 1.5	1222.52	1.00	0.00	0.00	0.00	0.00
> 1.5	521.14	0.00	1.00	0.00	0.00	0.00

從第一關到第十二關的概觀我就以折線圖的方式表示(如圖 5)，可以發現實驗組比對照組的結果好。由此可見，貝式網路的學習法對學習者的學習是有效的。

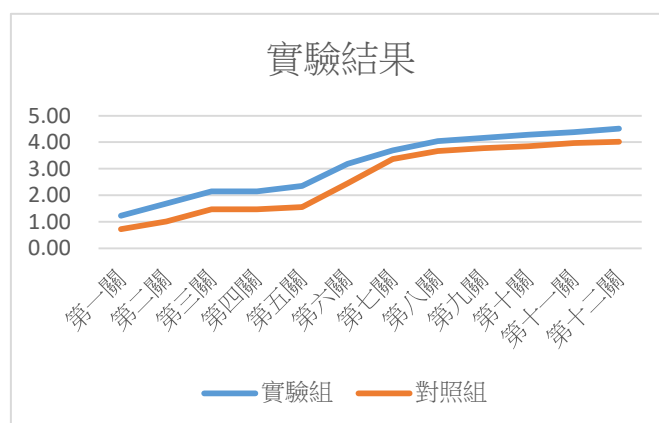


圖 5 實驗結果

第五章 結論與建議

第一節 結論

根據研究結果顯示，適性化的教學方式對於學習者的學習成效是有效的，雖在第九關以前皆無明顯的分數差距，但在第九關開始為複合題型，若是前面關卡的基礎概念沒有理解或是不甚清楚，則在第九關就可以看出學生的回答狀況的表現。因我們考慮在偏鄉教育在有限的資源情況下，使用了實體化的視覺化教材，來進行實驗，也就相對地，我們給的積木元件數量也是有限的，若是超出此積木元件的數量，也就代表此答案不是最佳解。經由這次的實驗結果發現到，越到後面關卡，對照組的學生會停留在此關卡的次數越多，解題時間也就越長，一直在同樣的關卡解一樣的題目，也不甚清楚自己錯在哪個概念，該使用哪塊積木元件解題。在實驗組，學生對於第九關之後的關卡，會停留在此關卡的次數越低，解題時間也會較短，分數也會越高，到達下一關卡的機會也就越高。顯然地，實驗組在前面關卡的所有概念皆能清楚，在第九關後的關卡，能知道要使用適當的積木元件去達到目標。結果顯示，以貝式網路為基礎的適性化教學對學習成效是有效的。

教育部將於 108 年課綱將「科技領域」加入為第八大領域的必修課，運算思維是屬於高層次思考的能力，必須經由生活經驗的累積與思考，才能有效地解決問題。因此透過科技領域的設立，將科技與工程之內涵納入科技領域之課程規劃，藉以強化學生的動手實作及跨學科，如科學、科技、工程、數學（Science, Technology, Engineering, and Mathematics, STEM）等知識整合運用的能力。（十二年國民基本教育課程綱要-國民中小學暨普通型高級中等學校, 105）。

程式語言因其學習的困難度相較於其他科目高出許多並且我們應該著重於「運算思維」而非程式語言的學習上。所以為了降低其學習困難度，本研究使用了 Code.org 作為其主要運算思維課程設計的訓練關卡，也考慮到偏鄉教育的資源缺乏，使用視覺化元件積木，提起學習者的興趣，也藉以降低學生學習新課程的排斥力，也可減少學習程式語言繁瑣的規矩與錯誤率。因此根據結果可知，「即時適性化架構運用於運算思維課程設計」對學習者是有效的。

參考文獻

1. 十二年國民基本教育課程綱要總綱(教育部發布版)(2016)。十二年國民基本教育課程綱要國民中小學暨普通型高級中等學校-科技領域。十二年國民基本教育課程綱要總綱(教育部發布版)，頁 5-24，https://www.naer.edu.tw/ezfiles/0/1000/attach/92/pta_10229_131308_94274.pdf。
2. 毛連塏&陳麗華 (1991)。精熟學習法。臺北市:心理。
3. 王昌斌,吳俊毅 & 楊惠媚 (2003)。整合式主動推播學習平台模型以課程知識為基礎。中原學報(人文及社會科學系列), 31(4)，頁 423-433。
4. 伍宏麟(2011)-教學策略與學習工具對程式語言初學者學習成效及學習態度之影響，頁 10-23。
5. 伍宏麟-教學策略與學習工具對程式語言初學者學習成效及學習態度之影響(全國論文網 100/06)
6. 余民寧 (1994)。學習評量與 SP 表分析。T&D 飛訊第 217 期，臺北市:臺灣書店，頁 4-6。
7. 李威霖 (2016)。情境式運算思維教材之發展與評估。頁 1-10。
8. 李威霖-情境式運算思維教材之發展與評估(全國論文網 105/06)
9. 林淑文 (2010)。探討國字筆順適性化教學基於試題反應理論和遊戲式學習之研究。
10. 林淑文 (2010)。探討國字筆順適性化教學基於試題反應理論和遊戲式學習之研究。國立虎尾科技大學資訊管理研究所碩士論文。
11. 孫扶志 - 動機與自我調節策略對國小學童學科學習之影響:https://lms.cit.cyut.edu.tw/sys/read_attach.php?id=2352160
12. 張素芬 (2010)。國小資訊教育實施 Scratch 軟體教學之研究。
13. 張皓程 (2017)。即時焦慮適性化提示策略於溝通式英語學習之成效，頁 5-10。
14. 莊惠淇 (2014)。中小學資訊科技課程運算思維內涵規劃。頁 5-9。
15. 陳俊德,何明亮 & 陳佑錠 (2006)。主題學習目標之個人化數位教材服務機制。資訊管理學術與實務研討會論文集，頁 325-332。
16. 陳胤丞 (2017)。探討不同先備知識學生使用程式教學遊戲之行為意圖，頁 1-7。
17. 陳胤丞-探討不同先備知識學生使用程式教學遊戲之行為意圖(全國論文網 106/01)
18. 黃國禎,蘇俊銘 & 陳年興 (2012)。數位學習理論與實務，頁 1-55。
19. 黃國豪,葉晟德,王士晉,陳碧茵,林春合 & 賴世偉 (2007)。結合多風格量表之適性化數位學習系統。理工研究學報，41(2)，頁 25-42。

20. 楊建民 (2010)。探究式教學法與講述式教學法在國小 Scratch 程式教學學習成效之研究，頁 20-22。
21. 廖怡雯 (2014)。以認知風格為基礎之適性化數位學習系統之成效—以國小自然科動物的繁殖和行為為例，頁 8-23。
22. 廖怡雯-以認知風格為基礎之適性化數位學習系統之成效—以國小自然科動物的繁殖和行為為例(全國論文網 103/06)
23. 廖述賢&溫志皓 (2011)。資料探勘理論與應用:以 IBM SPSS Modeler 為範例。博碩文化，頁第九章
24. 劉正吉(2011)-以 Scratch 同儕程式設計提升學童問題解決能力之探究，頁 9-35。
劉正吉-以 Scratch 同儕程式設計提升學童問題解決能力之探究(全國論文網 100/04)Peer instruction and Problem solving with Scratch programming <http://hdl.handle.net/11296/42mzr7>
25. 鄭宇琿(2012)-以 Scratch 結合樂高機器人在合作學習之探究-以國小高年級學生為例，頁 6-19。
26. 鄭宇琿-以 Scratch 結合樂高機器人在合作學習之探究-以國小高年級學生為例(全國論文網 101/06)
27. 親子天下 (2017)。108 新課綱系列 1 | 開學了，新課綱試行上路。國中小兩大改變正在發生，<https://flippedu.parenting.com.tw/article/3848>。
28. 謝錦泉 (2007)。以貝氏網路建構適性化學習環境。頁 8-15。
29. Antal, P., Verrelst, H., Timmerman, D., Huffel, S. V., Moor, B. d., & Vergote, I. (2000). Bayesian Networks in Ovarian Cancer Diagnosis: Potentials and Limitations. Paper presented at the 13th IEEE Symposium on Computer-Based Medical Systems.
30. Arroyo, I., & Woolf, B. P. (2005). Inferring learning and attitudes from a Bayesian Network of log file data. Paper presented at the 12th International Conference on Artificial Intelligence in Education.
31. Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2), p.e67-69.
32. Carroll, J. B. (1963). A model for school learning. *Teachers College Record.*, 64,p.e723-733.
33. Charniak, E. (1991).Bayesian networks without tears: making Bayesian networks more accessible to the probabilistically unsophisticated. *AI Magazine*, 12(4), p.e50-63.
34. Chen, C. M., Lee, H. M., & Chen, Y. H. (2005). Personalized e-learning system using item response theory, *Computers And Education.*, 44(3), p.e237-255.
35. Corney, D. (2000). Paper presented at the Evolutionary Design and

- Manufacture-ACDM 2000. *Designing food with bayesian belief networks*.
36. CSTA (2011). CSTA K-12 Computer Science Standards. The ACM K-12 EducationTask Force. Retrieved from http://www.csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf
 37. Dochy, F. (1992). Assessment of prior knowledge as a determinant for future learning: The use of prior knowledge state tests and knowledge profiles. Utrecht/London:Lemma/Jessica Kingsley.
 38. ECT. (2010). Google: Exploring Computational Thinking. Retrieved from <http://www.google.com/edu/computational-thinking/>
 39. Elawar, M. C., & Corno, L. (1985). A factorial experiment in teachers' written feedback on student homework: Changing teacher behavior a little rather than a lot., *Journal of Educational Psychology*., 77(2), p.e162-173.
 40. Facer, K. (2003). Screenplay: Children and computing in the home. *Psychology Press*.
 41. Gagne, R. M. W., Walter W;Golas, Katharine C;Keller, John M;Russell & James D. (2005). Principles of instructional design. *Performance Improvement*, 44(2), p.e44-46.
 42. Grover, S., & Pea, R. (2013). Computational Thinking in K-12 A Review of the State of the Field. *Educational Researcher*., 42(1), p.e38-43. <http://hdl.handle.net/11296/q3tned>
 43. Hung-Lin Wu-The Effects of Instructional Strategy and Learning Tool on Senior High Students' Programming Learning : <http://hdl.handle.net/11296/3n83fe>
 44. ISTE (2011). Operational Definition of Computational Thinking for K-12 Education. Retrieved from <https://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>
 45. Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*., 37(2), p.e83-137.
 46. Lu, J. J., & Fletcher, G. H. L. (2010). Thinking about computational thinking. *SIGCSE Bull.*, 41(1), p.e260-264. doi: 10.1145/1539024.1508959
 47. Moerkerke, G. (1996). Assessment for flexible learning. Utrecht: Lemma.
 48. Papoulis A. (1984). Probability, Random Variables, and Stochastic Processes, 2nd edition. Section 7.3. New York: McGraw-Hill.
 49. Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *ArtificialIntelligence*, 29(3), p.e241-288.
 50. Przytula, K. W., & Thompson, D. (2000). Construction of Bayesian networks for diagnostics.

51. Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM technical symposium on Computer science education.*, p.e265-269.
52. Wang, T.I., Wang, K.T., & Huang, Y.M. (2008). Using A Style-Based AntColony System for Adaptive Learning, *Expert Systems With Applications.*, 34(4), p.e2449-2464.
53. Weibelzahl, S. (2001). Evaluation of adaptive systems. In M. Bauer, P. J. Gmytrasiewicz, & J. Vassileva (Eds.), *User modeling: Proceedings of the eighth International conference*, UM2001 p.e 292-294. Berlin: Springer.
54. Wei-Lin Li-Teaching High School Students Computational Thinking with Situated Learning Materials.<http://hdl.handle.net/11296/pg36sj>
55. Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), p.e33-36.
56. Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions on the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), p.e3717-3725.
57. Wing, J. (2011). Research notebook: Computational thinking— What and why? TheLink Magazine, Spring. Carnegie Mellon University, Pittsburgh. Retrieved from <http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
58. Xenos, M. (2004). Prediction and assessment of student behaviour in open and distance education in computers using Bayesian networks. *Computers & Education*, 43,p.e345-359.
59. Yin-Cheng,Chen-To explore students with different prior knowledge in using coding games,their behavioral intentions.<http://hdl.handle.net/11296/cnb9q4>
60. Yi-Wen Liao-The Effects of Adaptive E-learning System Based on Cognitive Styles.<http://hdl.handle.net/11296/3yu6mn>
61. Yu Chun Cheng-A Case Study of Cooperative Learning that Integrated Scratch and LEGO WeDo for an Elementary School Students
62. Zheng, M., Spires, H. A., & Meluso, A. (2011). Examining upper elementary students' gameplay experience: A flow theory perspective. In A. Mendez-Vilas (Ed.), *Education in a technological world: Communicating current and emerging research and technological efforts.*, p.e190-198.

