

視覺化程式設計環境中之學習行為紀錄與分析

Learning Analytics in Visual Programming Environment

潘俐璇¹ 林丞輕² 張智凱³

PAN, LI-HSUAN¹ LIN, CHENG-ZHI² CHANG, CHIH-KAI³

¹ 國立臺南大學 數位學習科技學系 大學生

¹ National University of Tainan, Department of Information and Learning Technology,
Undergraduate student

E-mail : S10655005@stumail.nutn.edu.tw

² 國立臺南大學 數位學習科技學系 碩士生

¹ National University of Tainan, Department of Information and Learning Technology,
Graduate student

E-mail : M10855005@stumail.nutn.edu.tw

³ 國立臺南大學 數位學習科技學系 教授

² National University of Tainan, Department of Information and Learning Technology,
Professor

E-mail : chihkai@mail.nutn.edu.tw

摘要

在現今資訊教育普及的社會，學生擁有程式設計的能力逐漸為社會所重視。現在國小程式教育多使用視覺化程式語言 Scratch 進程式學習，當學童使用視覺化程式語言時，通常僅能看見最後專案結果，未能參考自我之撰寫程式過程，不僅無法針對本身學習上的困難進行反思與檢討，同儕間的交流也因不了解程式撰寫的步驟，僅能直接參照他人程式逕行模仿，無法達成互相學習之目的。然而目前尚未有人製作在視覺化程式語言的環境下之學習行為紀錄系統，因此，本計畫預期以 OpenCV 模組，開發一針對視覺化程式語言之學習行為紀錄系統，紀錄學生撰寫程式之過程，並產生學習歷程之紀錄檔。本計畫目的在紀錄學生之程式撰寫過程，並分析學生撰寫程式之特定行為。學生使用所開發之系統參照過去行為紀錄，讓學生能達成後設認知(Metacognition)之效，改善自身之學習；同儕間亦能互相參考、交流彼此撰寫程式流程，達到知識共享(Knowledge Sharing)之目的；教師藉由分析學生之學習行為紀錄，評估教學成效與品質，並作為改善教學之依據。本計畫期望培養學生可以帶得走的能力，善用資訊科技優勢，務實地實踐於現實生活。

關鍵字：程式設計、視覺化程式語言、學習行為分析、後設認知、知識共享

Abstract

Nowadays, in a society where computational education is popular, students' programming ability is gradually valued by society. Now, program education in primary school usually uses Scratch, visual programming language, for program learning. When students use visual programming language, they can usually only see the final project results, but fail to watch the self-programming process. Because the students don't understand the steps of programming, peers' communication could neither reflect and review for their learning difficulties nor achieve the purpose of mutual learning. They could only directly imitate others' programs. However, no one has yet produced a learning behavior recording system in the context of a visual programming language. Therefore, this project expects to use the OpenCV module to develop a learning behavior recording system, to record the students' programming process and generate a record file of the learning process. The purpose of this project is to record the student's programming process and analyze the specific behaviors while a student is programming. Students use this system to observe past behavior records so that students can achieve the effect of metacognition and improve their learning; peers can take each other as the references, communicating with others to achieve the purpose of knowledge

sharing. By analyzing the student's learning behavior records, teachers evaluate teaching effectiveness and quality and use it as the basis for improving teaching. This project hopes to build up the abilities for students to engrave on their minds, make full use of the advantages of computational technology, and pragmatically practice in real life.

Keywords: programming, visual programming language, learning analytics, metacognition, knowledge sharing

壹、前言

一、研究動機

數位化的時代中，教導學生如何運用資訊科技自我學習與解決問題，對教育而言是個重要的課題。除了提升學生的資訊素養，運用資訊科技學會閱讀、數學、科學等傳統學科之外，更重要的是讓學生學會運用電腦科學的思維與技巧來解決問題，充分發揮資訊科技的優勢。因此近年各先進國家無不大力提倡中小學資訊教育的改革，為取代傳統的程式語言，現一般中小學大多推廣使用視覺化程式語言(visual programming language)。這類的視覺化程式語言通常具有低門檻(low floor)、高效用(high ceiling)、具備學習遷移(enable learning transfer)等特性，知名的工具譬如 Scratch、Minecraft、Alice、AgentSheets 等。視覺化程式語言使用圖形化元素來進行程式設計，

利用圖形與文字的簡單排列，讓學習者能迅速地掌握整個程式架構、人性化地進行偵錯與程式維護，進而降低傳統程式語言帶來的認知負荷。目前在國內國小資訊教材中最常使用的視覺化程式語言為 Scratch，用以輔助學習程式設計，然而目前沒有針對 Scratch3.0 所開發之學習行為紀錄系統，學生完成專案後僅能看到最後的執行成果，卻看不到程式撰寫過程。即使學生想從過去撰寫過程中找出問題癥結點進行反思，仍不得其門而入；而同儕間互相交流、觀看特定專案時，只能看到執行結果，卻無法讀出他人的程式撰寫步驟及順序，亦無助於困惑之學生解決問題。因此，本計畫期望設計一學習行為紀錄系統，紀錄學生之程式撰寫歷程，不僅能幫助學生自我反思學習經過，同儕間也能互相觀摩參照對方之程式撰寫過程，教師亦能從中蒐集數據，藉以改良教學品質。

二、研究問題

根據上述的研究動機，本計畫重點在於開發一針對 Scratch3.0 視覺化程式環境之學習行為紀錄系統，期望透過此系統提升教師教學效率與學生學習成效，並探討以下研究問題：

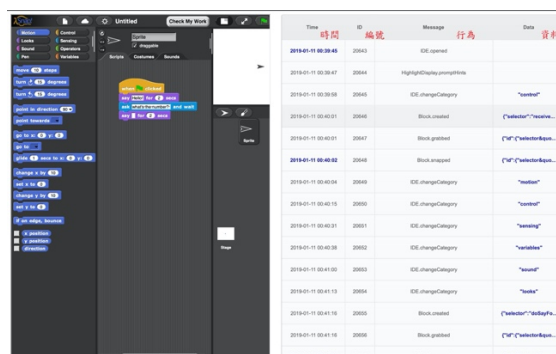
1. 如何開發一系統在 Scratch3.0 視覺化程式語言的環境下，詳細紀錄學生撰寫程式之過程與行為？
2. 系統如何彙整學生撰寫程式之過程與行為，並提供其他學生進行參考與學習？
3. 如何使學生透過本計畫開發之學習行為紀錄系統對自我的專案進行反思？

貳、 文獻回顧與探討

一、 學習行為及分析

(一) 學習行為紀錄

學習行為紀錄系統為根據學習分析(Learning Analytics)理論，所產生紀錄使用者行為之系統。目的在於提供使用者過去使用行為之紀錄，作為日後參考依據，便於修改、校正及分析；在教學的角度而言，將學習歷程當作評量學生的工具，是多數研究者提及的重點之一(伍柏翰，2017)。教師能藉由學生之行為紀錄，觀察分析學生學習行為，以改良教學品質。在諸多程式撰寫工具中，大多備有行為紀錄之工具，惟紀錄方法之多，以下為其中幾項舉例。iSnap 在紀錄資料集方面，提供一列表列出所有專案資料，包含專案編號、專案作業名稱、起始與結束時間、專案共有幾個紀錄點，可點選專案編號，即可進入特定專案中的詳細行為紀錄資料。而其行為紀錄資料可供他人查詢、觀看特定程式撰寫過程，iSnap 會依照時間順序排列成清單，包含時間(Time)、編號(ID)、行為(Message)、資料(Data)，如下圖 1。而 NetLogo 中，則分別紀錄了紀錄器、事件、階層。



Time	ID	Message	行為	Data
2019-01-11 09:29:45	20043	ESC opened		
2019-01-11 09:29:47	20044	HighlightDisplayProperties		
2019-01-11 09:29:58	20045	ESC changeCategory		
2019-01-11 09:30:21	20046	Block created		"Patcher" "highlightP...
2019-01-11 09:30:21	20047	Block grabbed		"Patcher" "highlightP...
2019-01-11 09:40:02	20048	Block snapped		"Patcher" "highlightP...
2019-01-11 09:40:04	20049	ESC changeCategory		
2019-01-11 09:40:15	20050	ESC changeCategory		
2019-01-11 09:40:31	20051	ESC changeCategory		
2019-01-11 09:40:38	20052	ESC changeCategory		
2019-01-11 09:41:00	20053	ESC changeCategory		
2019-01-11 09:41:13	20054	ESC changeCategory		
2019-01-11 09:41:16	20055	Block created		"Patcher" "highlightP...
2019-01-11 09:41:16	20056	Block grabbed		"Patcher" "highlightP...
2019-01-11 09:41:44	20057	Block snapped		"Patcher" "highlightP...

圖 1、iSnap 之學習行為紀錄系統紀錄內容參照表

(二) 學習行為分析 (Learning Analytics)

一般傳統教學中，所有學生須以同樣的學習速度學習一項新知識。然而，學生天生資質歧異，以同樣的學習速度，勢必造成不同水平之學生學習成果不均、教學成效不彰。對此，林奇賢、李玉順(2016)認為學習分析為學生提供適性化的教學，讓每位學生能有效地以各自的學習速度學習。Siemens, G.與 Gašević, D. (2012)將學習分析定義為為了理解和優化學習歷程及其發生的環境，測量、收集、分析，和報告有關學習者及其背景數據。學習分析的目的是幫助學校、教師，以及學習者本身，瞭解其學習經歷及環境，並加以改良學習行為。在優化學習的過程中，將收集而來的數據進行分析，並產生多項回饋，回饋結果能再次運用在學習歷程中，藉此改良學習行為。學習分析在深入了解大部分高等教育的難題中至關重要，更是教師、學習者、學校制定變革的基礎(Siemens, G. & Long, P., 2011)。不僅學習者能在往後的學習中日益改善學習行為，教師及學校也能從中萃取相關資訊，作為日後課程規劃的依據。為此，本計畫預

計開發一項學習行為紀錄系統，紀錄學生操作過程及動態，以利作更進一步之研究。

二、後設認知(Metacognition)

學生在學習過程中，若發生了未預期的意外，或執行過程不順利，抑或是結果不符合預期，會使得學生進入反思的過程，此即為後設認知(Metacognition)的第一步驟。Flavell, J. H. (1979)最早提出後設認知的概念，稱為「認知的認知」或「知識的知識」。後設認知是一段個人探討自我認知的歷程，當個體對於自我的求知過程能夠掌握、控制，並加以體驗及覺察，即對自我的認知歷程和結果進行自我覺知、監督、調整與評鑑的一種知識、經驗與能力，也就是比原來所認知者高出一層的認知，並能對發現的問題不斷地進行計畫、檢驗、監控、修訂、評估等策略，進一步幫助自己釐清並解決問題。Ambrose, S. A.、Bridges, M. W.、DiPietro, M.、Lovett, M. C.與Norman, M. K. (2010)將後設認知視為一個循環，如下圖 2，包含取得問題、評估自我的優勢與劣勢、計畫執行方法、使用策略、反思。學生在第一步中，會清楚地了解他們所需完成的項目；接著，評估自我的優勢與劣勢；第三步，對初步的執行方法形成一個雛形，能概略性地想出如何執行並完成；第四步，展開行動並使用策略以達成目標；最後，藉由監視流程之運行，反思及調整自我的策略，並產生一個改良後的方案，並重新進入取得課題的環節。故後設認知最重要的目的在於讓學生學會運用後設學習的概念，應用於往後真實生活中，運用策略、批判思

考及自我反思等技巧來解決問題，充分發揮並學以致用。而本計畫期望藉由開發一完整系統，讓學生能運用後設認知之能力，觀察並分析自己的學習歷程，並反覆進行思考，達到自我解決問題並改善學習方法之能力。

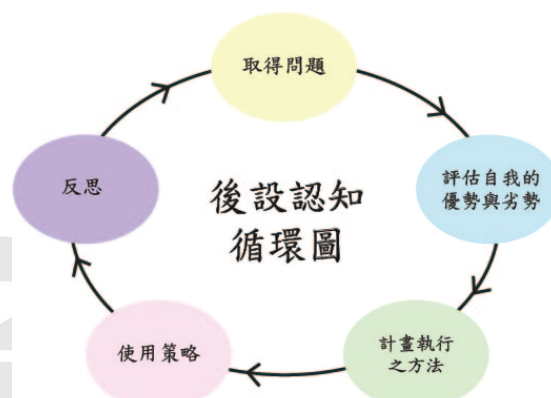


圖 2、後設認知循環圖

三、知識共享(Knowledge Sharing)

隨著科技日新月異的發展，知識的取得方式不再侷限於書本或學校，在資訊爆炸的時代，人人都可以透過任何管道取得知識。知識被視為組織中一項最重要的資源(Nahapiet, J. & Ghoshal, S., 1998; Spender, J.C. & Grant, R.M., 1996)，從組織的觀點而言，員工的表現可以藉由提供實用且相關的知識而提升(Alavi, M. & Leidner, D. E., 2001; Hansen, M. T., Nohria, N. & Tierney, T., 1999)。知識管理(Knowledge management)在近幾年曾被學術及實作領域廣泛討論及搜尋(Ismail, M. & Chua, L.Y., 2005)。Tiwana, A.(2002)將知識管理分為三個不同的過程：知識取得(Knowledge Acquisition)、知識分享(Knowledge Sharing)，以及知識運用(Knowledge Utilisation)。知識共享是知識管理的關

鍵，指組織的員工或內外部團隊在組織內部或跨組織之間，彼此透過各種渠道進行知識交換和討論，其目的在於透過知識的交流，擴大知識的利用價值並產生知識的效應(樊治平、孫永洪，2006)。因此，知識共享最重要的即是「交換經驗」，將自身對某事的理解、想法，或一時興起的好奇心，能透過互惠得到更多見解和認識。例如：GitHub 讓創作者將自己的專案放到平台上，除了可以儲存各種語言、各個版本的專案，同時也分享給其他使用者參考，創作者因此可以推廣專案，也能接收到其他使用者的意見、指教，使用者也能藉由參考去探索不同的技術，任何人都能在平台上互相切磋交流，並達到知識共享的目的；又如線上編輯器 Codepen，除了可以直接將連結貼給他人以觀看原始碼，還能搜尋並蒐藏他人的原始碼。學習者在學習一項新知識時，不免會產生疑問及困惑，若能發揮知識共享，透過同儕年齡相仿，思維水準接近，相互理解協助、指導、問答等，讓學習成效更為提升(李建億、胡政文，2018)。同儕間互相交流、觀摩對方撰寫的過程，排除問題所在，並汲取他人經驗，勢必能大幅提升學習成效。本計畫期望藉由開發學習行為紀錄系統，輔助學生互相觀摩借鏡，以利學習。

四、圖形導向計算(Picture-Driven Computing)

一般程式語言在學習過程中，學習者需記憶各個指令的作用，並了解如何編排才能順利運行。當學習者遇到程式錯誤而不能順利除錯(Debug)時，就很容易感到困難而放棄。然而，圖形導向計算(Picture-Driven

Computing)大幅降低程式語言的學習難度，藉由電腦上擷取螢幕畫面的功能，將欲控制的物件擷取成圖像，並撰寫所需的程式指令，進而達到控制物件的效果。例如，麻省理工學院所開發的 Sikuli 正是具有圖形導向計算特色的程式開發工具，並藉由圖形化使用者介面(Graphical User Interface, GUI)，讓使用者對自己的電腦畫面進行偵測和分析。透過 GUI 的使用，大幅降低傳統程式語言設計中，使用繁瑣的語法或是複雜的從屬關係的門檻，學習者不必逐行撰寫程式碼，僅需按下按鈕即可完成撰寫指令，並適當地插入物件圖像，就能完成所有程式碼，以自動操控整個螢幕畫面。本計畫係以 OpenCV 進行圖形導向計算實作，作為本系統開發之根據。

參、研究方法及設計

一、研究工具——Scratch3.0

Scratch 由美國麻省理工學院媒體實驗室所開發出包含豐富媒體庫的視覺化程式語言，是一種可將互動設計遊戲、故事動畫以及多媒體專案應用於課堂上的新技術(Brennan, K. & Resnick, M., 2013)。Scratch 視覺化程式語言透過操作程式圖塊與流程圖的方式，將抽象的概念轉化為圖像可以幫助學生提升偵錯能力、觀察變量與邏輯能力，並使其能修正程式、解決問題。Scratch 就像是組合玩具一般，透過組合即可將程式圖塊組裝成視覺化程式，學習者操作視覺化介面時，可在設計程式的過程中建立基本概念，並

透過幾何、計算、統計等，數學概念作為解決問題的關鍵，並營造一個快樂的學習環境，以培養學習者的創造性思維(He, Y. Y., Chang, C. K., & Liu, B. J., 2010; Lee, Y. J., 2011)。

根據 Scratch 官方網站公告於 2018 年 8 月開放 Scratch 3.0 Beta 版，預計於 2019 年 1 月 2 日正式版上線 Scratch 3.0 是 MIT 與 Google 合作，採用 HTML5 所開發的，不再依賴 Flash，且可相容於 Scratch 2.0，操作介面區域有大幅調整，程式圖塊也有增加一些新圖塊。Scratch3.0 介面分為五個區域，如下圖 3 所示，執行結果區、舞台區、角色區、程式圖塊選取區和程式編輯區。而程式圖塊選取區也根據程式設計概念，將所有圖塊以顏色分出十種類別為：動作、外觀、音效、畫筆、資料、事件、控制、偵測、運算、更多圖塊。撰寫程式僅需要透過拖曳的方式將圖塊由程式圖塊選取區中移至程式編輯區，而每個角色都有一個屬於自己的程式編輯區域，可在指定的時間內執行許多不同角色的程式。另外，將程式圖塊選取區的更多圖塊類別分成函式圖塊與擴充功能，並將畫筆、視訊和音效類別的圖塊放置於擴充功能裡，讓學習者根據需求再添加，而資料類別圖塊則改名為變數類別，最特別的是在介面左下方的添加擴展，可以新增其他擴充功能，例如：音樂、畫筆、視訊偵測、文字轉語音、翻譯、以及 LEGO WeDo 2.0 等。透過將傳統的文字程式語言視覺化，並利用圖塊玩具的原理拖曳、組裝，以取代打字建立程式的方法，除了可避免學習者指令輸入錯誤外，也可更快速、直觀、輕鬆地學習。



圖 3、Scratch3.0 介面圖

二、開發工具——OpenCV

OpenCV(Open Source Computer Vision Library)為一個專門處理電腦視覺的開源函式庫，主要是由 Intel 公司發起與開發，以 BSD 授權條款發行，可以在商業或是研究領域中免費使用。OpenCV 專案最早於 1999 年啟動，起初的主要目的有以下三個：一、透過提供開源且最佳化的基礎函式庫，以推進機器視覺的研究；二、提供共同的基礎函式庫使得知識傳播更容易；三、透過提供授權促進商業應用軟體的開發。OpenCV 主要使用 C++ 程式語言進行開發，因此大部分的主要程式介面由 C++ 所組成，只有少部分的 C 語言介面。然而，為了給予廣大的使用者廣泛使用，OpenCV 也提供其他不同程式語言供使用者使用，包含 Python、Java、MATLAB、C#、JavaScript 等。OpenCV 主要應用在影像處理，函式庫裡內建許多有關影像處理的演算法，同時函式庫裡也包含不同的影像處理模組。OpenCV 可利用讀寫圖片與影片、矩陣運算、統計、影像處理等功能，處理物體追蹤、人臉辨識、紋理分析、動態視訊的影像處理等問題。物體追蹤會透過影像變化找到影像中正在移動的物

體；人臉辨識針對臉的特徵找到圖片或是影片中人臉所在的位置；紋理分析常應用於醫學 X 光片、細胞圖像判讀、以及航空地形照片分析。因此，本計畫在視覺化程式設計環境中，使用 OpenCV 為開發學習行為紀錄系統的主要工具。

三、 開發流程設計

本計畫預計開發一基於 OpenCV 函式庫的視覺化程式設計環境學習行為紀錄系統，如下圖 4 所示。在學習者操作 Scratch 時，行為蒐集模組會先將整個介面透過程式做概略性的劃分並偵測，分為圖塊(block)、程式編輯區(workspace)、舞台(stage)。當學習者開始操作時，系統會先依照劃分區域固定一段時間擷取圖片，之後隨即啟動 OpenCV 函式庫進行影像辨識，將所當時所擷取的圖片與最後一次被紀錄下的圖片進行比對。若有不同之處，則進一步分析、紀錄該圖片中學習者的行為，其行為可能包含點擊程式圖塊、拖動程式圖塊或輸入文字等。系統將這些行為資料轉換為文字型式紀錄學習者之行為，並將詳細資料存進行為資料庫中，再透過資料前處理模組，針對資料進行彙整與修改。最後，透過資料視覺化模組導出行為紀錄表給予教師或學習者觀看、查詢，而紀錄項目與格式將參考 iSnap 的行為紀錄表，因此預期行為紀錄表中會包含專案編號、行為編號、行為類型、行為內容、照提示操作、時間戳記，可從中得知系統所蒐集之行為參數，除了可得知何時進行何種行為，也可將行為紀錄分享給其他學生觀看，藉以互相觀摩學習。

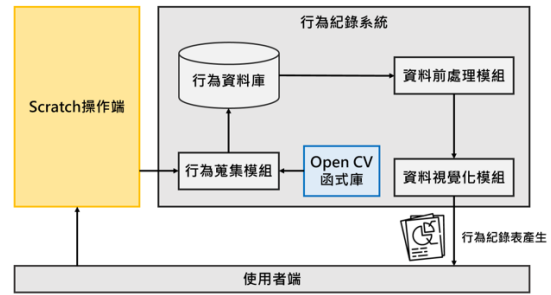


圖 4、視覺化程式設計環境之學習行為紀錄系統流程圖

肆、 研究結果

一、 系統開發流程

本計畫以 OpenCV 函式庫作為系統開發工具，在視覺化程式設計環境之下進行學習行為紀錄，如圖 5，並使用圖形導向計算作為系統開發之主要依據。在學習者開始操作 Scratch 時，系統首先以 Scratch 中程式圖塊選取區之「圖塊類別」作為標的，啟動 OpenCV 函式庫進行影像辨識，當該類圖塊被選取時，類別的背景色會由白色轉為灰色，由此得知當前使用者所使用的圖塊類別。系統藉由不斷地偵測該區塊，一旦學習者切換所使用的圖塊類別時，即刻進行資料儲存及分析。系統會自動擷取程式編輯區之螢幕畫面，分別進行「圖片存檔」及「圖塊分析」：前者將螢幕擷取畫面以時間戳記為檔名存檔，供使用者依序查閱組合圖塊的過程及變化；後者將當時所擷取的螢幕畫面從圖塊邊緣剖析，畫出所有圖塊的輪廓，並計算每個顏色所涵蓋的面積，以此紀錄各類圖塊的使用情形。系統將這些使用者的資料數據儲存至資料庫中，再透過資料視覺化模組導出圖塊使用情況分析表給予教師

或學習者觀看、查詢，可從中得知系統所蒐集之行為參數，不僅能讓學習者了解自己所使用的圖塊概況、反思自己的學習行為；亦能輔助教師理解學習者的學習動態，以此輔助改善教學品質；同儕間也能互相觀摩、交流撰寫過程，提升學習者之學習成效。

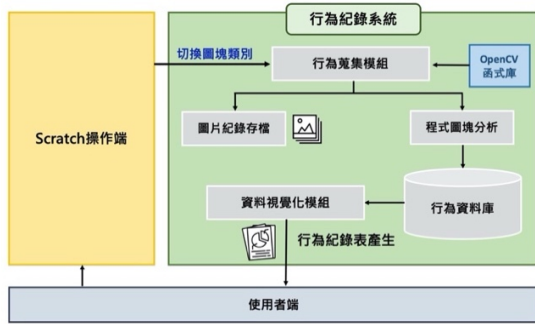


圖 5、視覺化程式設計環境之學習行為紀錄系統流程圖

車遊戲、敲磚塊等，皆有許多不同的使用者進行創作、改編等。本計畫以經典遊戲專案——敲磚塊為例，由使用者 4computer 製作，如下圖 7。該遊戲的遊玩過程簡易好上手，玩家負責以滑鼠左右操控一平板，並在球多次碰撞過程中，設法以平板接住球而不讓球落地，球便會再次彈起敲擊上方的磚塊，擊中越多磚塊的玩家分數越高。

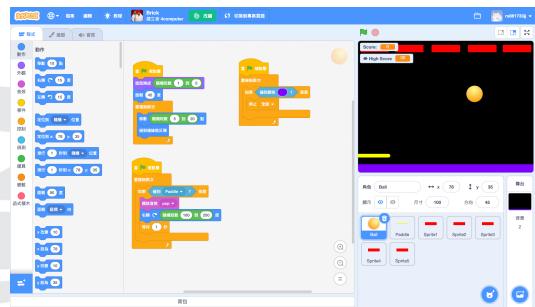


圖 7、敲磚塊專案畫面

系統執行成果如下圖 6，在切換圖塊的瞬間，分析各類圖塊的面積比例，並將資料儲存至行為資料庫中。



圖 6、系統執行之成果說明

二、經典專案分析——以敲磚塊為例

Scratch 官方網站上富有大量多元且有趣的專案，創作者能夠將自己的專案透過「分享」的方式，公開給所有使用者任意改編及操作，為知識共享奠下良好的基礎。其中，不少使用者善於以 Scratch 進行遊戲設計，舉凡經典小遊戲：貪食蛇、吃豆人(Pac-man)、賽

本計畫針對遊戲中到處彈跳、碰撞的「球」進程式圖塊分析，根據各類圖塊需使用的比例繪製出目標折線。並邀請國立臺南大學數位學習科技學系，正在學習程式設計之大一學生進行系統測試，將該位學生撰寫程式過程中，每次擷取螢幕畫面時所使用的圖塊進行分析，同樣會製成折線圖加以比對，如圖 8。藉此視覺化資料提供教師檢視、觀察學習者的學習情況，發現學生撰寫程式時可能遭遇的問題及盲點，適時提供引導，進而提升學習之最佳效益。

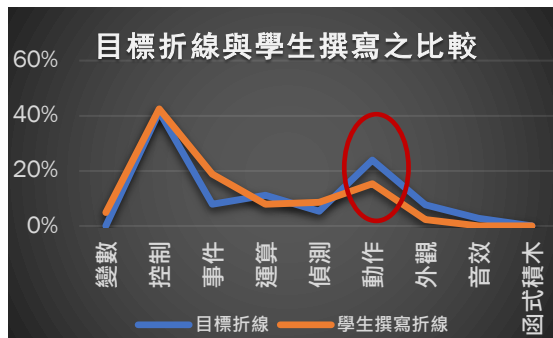


圖 8、目標與學生撰寫程式比較圖

三、後設認知思維與圖塊變化——以動作圖塊為例

從圖 8 中可以得知，學生在撰寫程式過程中與專案完成目標之差距，並針對落差較大之「動作圖塊」進行分析。將開始撰寫專案至結束中的所有變化繪製成折線圖，觀察學生使用單一圖塊的使用情形，如圖 9。開始撰寫程式時，學生較容易從「事件」圖塊著手，因此初始值為 0%；有了觸發條件之後，學生便容易陷入大量使用動作圖塊的思考中，由於動作圖塊直觀易懂，且學習者能立即看到成效，因此多數學習者皆仰賴動作圖塊完成大部分的指令。然而，隨著撰寫專案之時間增加，學生亦會發現使用過多動作圖塊的不便之處，因此會逐漸添加其他指令，取代多而繁複的動作圖塊，逐步逼近動作圖塊之目標折線。學習者在撰寫程式的過程難以一次到位，因此在過程中會展現曲折之變化，表示學生不停地在增加與刪除中取捨，一步一步將程式完整。因此根據整體撰寫趨勢而言，隨著時間推進，學生之撰寫折線會益發靠近目標折線。在不斷地修改與思考的過程中，學生一再地取得問題、評估自己所能採取的方法，並嘗試將之實現，應用後設認知的方法學習程式設計，在每次試驗中重新整理

思路，進一步從自我的學習歷程中反覆探索與進步。

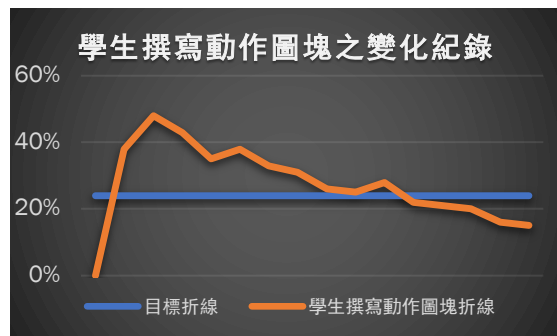


圖 9、學生撰寫動作圖塊之變化與目標折線比較圖

四、未來展望

惟計畫執行過程中，難免遇到室礙難行的困難，因而留下些許遺憾。原訂計畫中希望能準確辨識出學習者的使用行為，包含點擊程式圖塊、拖動程式圖塊或輸入文字等，然而辨識技術尚不夠嫻熟，最後僅能選擇以圖塊類別作為辨識標的，並擷取畫面分析圖塊之使用結果。本計畫期許未來能改善以下三項缺失：其一為倘若使用者固定使用同一類圖塊而未切換其他類別，極可能僅記錄第一秒之程式，而遺失後續拼接、組裝之過程。其二為由於系統著重於程式圖塊選取區及程式編輯區，若學習者在學習過程中，交叉製作其他角色之程式圖塊，系統辨識便因此失效且不準確，亦是本系統之疏漏。其三為系統無法即時展現學習者之學習動態變化，統計數據為輸出後再行視覺化，擷取之圖像亦僅保存於學習者所使用的電腦中，因此希望能在系統中額外添加圖片瀏覽器的視窗，依序匯入擷取之圖片，以提供學習者隨時能瀏覽過去之程式編輯過程。最後，本系統亦期望未來能增添推薦

功能，輔助學生間相互學習，達成知識共享的目的。

伍、 結論

由於資訊科技的蓬勃發展，將教育結合科技以改良傳統教學，成為現代社會中的一股趨勢。高科技的發展帶動資訊人才的需求，因此程式設計納入中小學的課綱中，成為人人都必須擁有的技能。對於初學者而言，藉由視覺化程式語言學習程式觀念，能使程式設計更容易理解及操作，達到最佳學習成效。

本計畫藉由開發學習行為紀錄系統，讓學習者之學習歷程可視化，學生能看見專案從無到有，一步一步地建置過程，進一步監督及反思，回憶當時撰寫程式時所疑問或無法解決的難題，針對問題點改善學習行為、理解解決問題之方法；同儕間也能夠互相交流，根據學習歷程討論、分析，在教學相長之際共同成長茁壯。除此之外，本計劃期許培養學生在錯誤中學習的成長型思維，能檢視錯誤及問題，將之視為通往成功的必經之路而不挫折，重新反思內化後將學習成效最大化。期許本系統之開發能落實輔助學習者學習之目的，教師根據系統分析結果提升學生學習環境，改善資訊教育之品質與成效，有效地幫助學生學習解決問題及後設認知的技能，以達到自律學習之目的。

致謝

本計畫感謝科技部補助大專學生研究計畫之成果經費，大專生計畫編號 MOST 108-2813-C-024-009-H；以及

科技部補助專題研究計畫，計畫編號 MOST 105-2511-S-024 -007 -MY3，特此感謝。

參考文獻

一、 中文部分

伍柏翰 (2017)。結合即時診斷機制的行動學習與自我調整學習模式對學生運用概念圖的學習成就與行為之影響。*數位學習科技期刊*，9 (2)，1-27。

李建億、胡政文 (2018)。在課堂中運用自動動態分組於合作學習以協助學習困難之學生。*國立臺灣科技大學人文社會學報*，14 (2)，117-137。

林奇賢、李玉順 (2016)。海峽兩岸國民小學學生之網路學習行為比較研究。*數位學習科技期刊*，8(3)，71-102。

樊治平、孫永洪 (2006)。知識共享研究綜述。*管理學報*，3 (3)，371-378。

二、 英文部分

Alavi, M. and Leidner, D. (2001), "Knowledge management and knowledge management systems: conceptual foundations and research issues", *Management Information Systems Quarterly*, Vol. 25 No. 1, pp. 107-36.

Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., & Norman, M. K. (2010). *How learning works: Seven research-*

- based principles for smart teaching. John Wiley & Sons.
- Brennan, K., & Resnick, M. (2013). Imagining, creating, playing, sharing, reflecting: How online community supports young people as designers of interactive media. In *Emerging Technologies for the Classroom* (pp. 253-268). Springer New York.
- Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *American psychologist*, 34(10), 906.
- Hansen, M. T., Nohria, N., & Tierney, T. (1999). What's your strategy for managing knowledge. *The knowledge management yearbook 2000-2001*, 77(2), 106-116.
- He, Y. Y., Chang, C. K., & Liu, B. J. (2010). Teaching Computer Programming for Freshmen: A Study on Using Scratch as Remedial Teaching. *International Journal on Digital Learning Technology*, 2(1), 11-32.
- Ismail, M. and Chua, L.Y. (2005), "Implication of knowledge management (KM) in higher learning institutions", paper presented at International Conference on Knowledge Management, PWTC, Kuala Lumpur.
- Lee, Y. J. (2011). *Scratch: Multimedia Programming Environment for Young Gifted Learners*. *Gifted Child Today*, 34(2), 26-31.
- Nahapiet, J. and Ghoshal, S. (1998), "Social capital, intellectual capital and the organisational advantage", *Academy of Management Review*, Vol. 23 No. 3, pp. 242-66.
- Spender, J.C. and Grant, R.M. (1996), "Knowledge and the firm: overview", *Strategic Management Journal*, Vol. 17, pp. 5-9.
- Siemens, G., & Gašević, D. (2012). Special Issue on Learning and Knowledge Analytics. *Educational Technology & Society*, 15(3), 1-163.
- Siemens, G., & Long, P. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE review*, 46(5), 30.
- Tiwana, A. (2002), *The Knowledge Management Toolkit: Orchestrating IT, Strategy and Knowledge Platforms*, Prentice-Hall, Englewood Cliffs, NJ.