

## 以 Greenfoot 提升高中生程式設計之學習態度

### Promoting the Programing Learning Attitude of Senior High School Students

#### Using Visualized Greenfoot Environment

黃雅琪、賴阿福

臺北市立大學 資訊科學系

sprout1019@gmail.com, laiahfur@gmail.com

**【摘要】** 程式設計學習對於初學者是充滿挑戰且複雜的任務，包含邏輯推理、語法與語意、程式架構與物件觀念等，視覺化的程式環境可以協助初學者解決學習程式的困境，因此本研究採用 Greenfoot 協助高中生學習程式之概念，進行一學期教學實驗，課程結束後，檢測學生對於程式設計之學習態度及物件概念之學習成就，研究結果顯示學生對於程式設計學習態度為正向積極，但由於時間過於短促，學生對於程式概念的熟悉度為中等。

**【關鍵字】** 程式設計；高中；Greenfoot；學習態度

*Abstract: Programming learning can promote the learners' high order thinking skills including computational thinking. However, computer programming is a complex task for novice due to its syntax and abstract concepts. Visualized programming environment can reduce the learning dilemma, such as Scratch, App inventor, and Greenfoot. As a result, the main purpose of this study is to investigate the impact of visualized and game-based programming environment on high school students' learning attitude and achievement by adopting Greenfoot. The participants are 32 senior high school students. The learning experiment lasted for one semester. The findings reveal that the student show high attitude towards programming learning, but their learning achievement is medium due to short time learning.*

**Keywords:** programming learning, senior high school, Greenfoot, learning attitude

## 1. 前言

藉由日常生活問題引發學生學習電腦興趣，建立「資訊科學基礎知識」和強化「邏輯思維及運用電腦解決問題能力」，是政府的重要教育政策（教育部，2009）。近年大學推甄成為高中進入大學的重要管道之一，不只在乎學生的考試成績，學生的邏輯思維及運用電腦解決問題之表現，成為重要升學條件之一。Valente（1994）指出程式設計分四循環步驟描述（description）、執行（execution）、反思（reflection）與除錯（debugging），當學生在撰寫程式過程中，需思考解決方法及解題的策略，並了解程式的概念與技巧，此歷程有助於問題解決能力的提升。學習程式設計並不是容易的事，然而使用視覺化 Greenfoot 軟體，以遊戲案例化的方式將能減少學習困境，因此本研究將採用此工具進行教學實驗，以探討其對高中生程式設計學習的影響。

## 2. 研究目的

為了幫助初學者能更有效學習程式設計，許多學者認為視覺化環境有助於程式設計的學習（賴阿福，2011），能降低程式設計的學習門檻，且能提升學習興趣（Kelleher & Pausch, 2005）。本研究主要目的在於探討 Greenfoot 對於中學生程式學習態度之影響及學習成就之影響。

### 3. 相關研究

程式設計初學者需先學習語法、語意與程式的結構，接著將知識轉換成演算法，最後編撰成程式碼，這些都是困難 (Brusilovskay et al.,1997 ; Jenkins, 2002)。接著初學者在學習程式語言中，會覺得抽象概念難以思考，且課堂的範例大多以傳統數學問題為主，更易造成學生學習恐懼 ( Jenkins, 2002 ; West & Rose, 2002)。為了改善學習情況，可在教學上以程式區塊方式講解，循序地引導學生思考，最後轉換成程式碼 (蔡育融， 2013)。Papert (1980) 認為程式語言應先降低學習門檻 (low floor)，提昇初學者學習意願，其後再讓初學者寫出較為困難的程式 (high ceiling)，最後讓不同學習風格和興趣者能夠意願使用 (wide walls) 不同類型的專案。但要同時滿足此三者特性較為困難，而 Keller 和 Pausch (2005) 指出視覺環境有三個共通點，簡化程式設計過程、提供學習者支援、引發學生學習動機。換言之，程式設計工具及環境會影響學習意願及成效，故本研究選用視覺化 Greenfoot，以進程式設計學習之實驗。

### 4. 研究方法

#### 4.1. 研究對象

在台北市某公立高中社團課程進行學習實驗，參與學生人數共 32 名，學生背景能力不同，因為社團課程之學習者包含高一和高二學生，該班級學生大多沒有上過相關程式設計的課程，因此假設該班級皆為程式設計初學者。

#### 4.2. 程式活動設計

本實驗採用英國肯特大學所開發自由軟體 Greenfoot (<http://www.greenfoot.org>) 協助高中生學習物件、分支、迴圈等基本概念，活動主要參考高慧君老師範例，以貓狗追逐例子 (圖 1)，首先先讓學生知道程式中的物件，分別對它們進行行為與屬性設定(高慧君，2012)。並探索 Java 的分支概念 if, else if, else 等分支概念，以進行判斷式撰寫。以射飛碟(圖 2)產生 2x7 飛碟物件，用來了解 for 迴圈;以燈泡控制開、關 (圖 3)，了解 array 與 arrayList 區別，最後給予學生進階練習設計，撰寫打磚塊之程式之小專題 (圖 4)。

	
<p>圖 1 貓狗追逐進階遊戲(修改自高慧君)</p>	<p>圖 2 射飛碟(修改自高慧君)</p>
	
<p>圖 3 開關燈遊戲(修改自高慧君)</p>	<p>圖 4 打磚塊改多分數功能(修改自高慧君)</p>

#### 4.3. 研究工具

本研究所採用工具分為兩種 (1) 程式設計之學習態度以及對於自我感知程式概念熟悉度: 採用李克特氏五等量表，由「完全同意」至「非常不同意」，其信度分析結果分別為學生學習態度之向度 Cronbach $\alpha$ = 0.899 及學生對程式的熟悉度之向度 Cronbach $\alpha$ = 0.784，顯示兩者都有不錯的信度。(2)程式設計成就測驗:依照課程的教材設計測驗卷，讓學生檢測是否了解教材內容，題目類型含有單選、複選、填充與程式題，內容包含物件、分支、變數等。

## 5. 研究結果與討論

從表 1 顯示知道，學習態度各態度平均數皆大於 3.22，介於普通及同意等級之間，其中以「學習 greenfoot 之程式設計有助於建立物件的概念」、「學習程式設計對於未來就業有幫助」與「只要時間許可，我一定可以把程式學好」都大於 4，表示 Greenfoot 對於學習有助益，程式設計學習具有用性，學生具有良好程式設計學習自我效能(self-learning efficacy)。

表 1 學生的學習態度

題目	平均數	標準差
學習 greenfoot 之程式設計有助於建立物件的概念	4.06	.669
學了 greenfoot 讓我更想學習程式設計技巧	3.56	.716
學了 greenfoot 讓我覺得程式設計很有趣	3.63	.833
我覺得學習程式設計對於數學有幫助	3.31	1.09
我覺得學習程式設計，可以讓我做事方法更有條理	3.72	.958
我覺得學習程式設計，可以讓我做事方法更有信心	3.25	.950
我覺得學習程式設計，可以讓我學習電腦更有興趣	3.69	.965
學習程式設計對於其他學科有幫助	3.53	1.02
學習程式設計對於未來升學有幫助	3.94	.982
學習程式設計對於未來就業有幫助	4.22	.906
當我撰寫程式中，經常讓我忘了時間已過很久	3.63	1.07
只要時間許可，我會想逐漸挑戰更難的程式	3.63	1.07
只要時間許可，我一定可以把程式學好	4.03	.967
寫完/修改好程式會讓我有成就感	3.91	.856

如表 2 顯示高中生對於程式概念，熟悉度都大於 3，介於普通及熟悉等級之間，可知學生對於程式中物件的概念與程式之變數等概念較為熟悉，簡言之，在 Greenfoot 中可以有效協助學生了解物件與變數觀念。

表 2 學生對程式的熟悉程度

題目	平均數	標準差
偵測是有困難的	3.37	1.100
熟悉程式語法是困難的	3.28	1.143
熟悉程式中物件的概念	3.53	1.047
熟悉程式中變數的概念	3.50	1.078
熟悉程式中陣列的概念	3.16	1.167
熟悉程式中for迴圈的概念	3.25	1.136
熟悉程式中while迴圈的概念	3.19	1.120
熟悉程式中函數或方法的概念	3.09	1.058
熟悉程式的編譯過程	3.03	1.062

由圖 5 得知高一的答對率比高二得來的多，代表高一在程式學習成就就較佳，但第九題與第十題撰寫分支及迴圈之程式，高二的答對率比高一來的高，意味著高二對於程式的邏輯擁有較佳的基礎，學生對於 Java 特性與變數的類型有不錯的認知表現，整體而言，學生在繼承的概念物件宣告迴圈及分支等概念有待加強，可藉由後續課程加以強化，透過程式仿作、修改、追蹤及偵錯、自行創作程式或解題等，建立正確概念及心智模式。

Wu, Y.-T., Chang, M., Li, B., Chan, T.-W., Kong, S. C., Lin, H.-C.-K., Chu, H.-C., Jan, M., Lee, M.-H., Dong, Y., Tse, K. H., Wong, T. L., & Li, P. (Eds.). (2016). *Conference Proceedings of the 20th Global Chinese Conference on Computers in Education 2016*. Hong Kong: The Hong Kong Institute of Education.

概念	題行	題目	高一答對率	高二答對率	平均答對率
繼承	單選	1.在Greenfoot中，下列哪一張圖為繼承的概念	0.06	0.04	0.23
特性	複選	2.下列哪些是Java語言的特性	0.74	0.69	0.72
型態	複選	3.在下列方法(method)中，哪些不會有回傳值？	0.71	0.65	0.68
型態	複選	4.在Greenfoot中，下列哪些變數(variable)是整數型態	0.77	0.72	0.75
迴圈	複選	5.在for迴圈結構: for(a;b;c){d}中，下列說明哪些是正確的？	0.64	0.68	0.66
陣列	複選	6.下列有關陣列(array)概念的描述哪些正確？	0.54	0.49	0.52
物件	填空	7.如需要請建立一個類別為Animal、名稱為cat的物件實體則指令該如何寫？	0.15	0.03	0.09
陣列	填空	8.判斷以下程式執行結果為何？	0.18	0.13	0.16
分支	程式	9.利用if、else if、else寫判斷式。	0.29	0.31	0.3
迴圈	程式	10.利用for迴圈寫以下程式。	0.11	0.23	0.17

圖 5 高一、高二在程式概念測驗之答對率

## 6. 研究結論與未來展望

由於高中生的功課壓力相當大，因此學生只能利用短暫時間進行程式設計學習，在本研究的高中生藉由社團時間進行程式設計概念及實作技巧學習，且在 Greenfoot 之視覺化學習環境中學習物件、分支、迴圈等基本概念。本次實驗結果顯示高中生之學習態度皆為正向積極，尤其是 Greenfoot 對於學習物件概念之幫助，程式對於未來工作之助益及學習自我效能，但在程式概念熟悉度因受學習時間因素影響，表現平平；未來實驗將增強實作練習機會，甚至結合其他學科進行跨領域之專題設計，以強化其程式設計及運算思維能力。

## 參考文獻

- 教育部 (2009)。普通高級中學資訊科課程綱要補充說明。臺北：教育部。
- 蔡育融 (2013)。應用 App Inventor 於高中程式設計教學之個案研究。台北市：國立師範大學。
- 高慧君 (2012)。Greenfoot 入門課程研習教材。台北市：臺北市立教育大學。
- Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., & Miller, P. (1997). Mini language: A way to learn programming principle. *Education and Information Technologies*, 2(1), 65-83.
- Jenkins, T. (2002). On the difficulty of learning to program. *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Science*, 53-58.
- Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, 37(2), 83-137.
- Lai, A. F. (2011). The learning effect of visualized programming for Noice. *GCCCE 2011*.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. NY, USA: Basic Books Inc.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). *Scratch: Programming for All*. *Communications of the ACM*, 52(11), 60-67.
- Valente, J. (1995). Logo as a window into the mind. *Logo Update*, 4(1). Retrieved from <http://el.media.mit.edu/logo-foundation/pubs/logoupdate/v4n1.html>
- West, M., & Ross, S. (2002). Retaining females in computer science: A new look at a persistent problem. *JCSC*, 17(5), 1-7.