

# Exploring the Effects of Visualization Tool on Learning to Program

## 視覺化工具融入程式語言教學之效益探討

Ming-Puu Chen, 陳明溥

Graduate Institute of Information & Computer Education  
National Taiwan Normal University  
Taipei, Taiwan  
mpchen@ntnu.edu.tw

Ku-Chou Tai, 戴谷州

Graduate Institute of Information & Computer Education  
National Taiwan Normal University  
Taipei, Taiwan  
ken.java@gmail.com

Nian-Shing Chen, 陳年興

Department of Information Management  
National Sun Yat-San University  
Kaohsiung, Taiwan  
nschen@mis.nsysu.edu.tw

**Abstract**—This study examined the effects of visualization tool and prior knowledge on novices' learning to program and attitudes. 79 freshmen, aged from 16 to 20 years old, participated in a 3-week 'For Loop' programming course unit. A visualization tool, *Jeliot 3*, was employed to assist novices to learn Java language. The results revealed that (a) learners with high prior knowledge outperformed those with low prior knowledge, (b) for the high prior knowledge learners, the visualization group revealed more positive attitude than the contrast group, and (c) the visualization group showed more positive attitude in the aspects of learning satisfaction, learning support and active learning and thinking.

**Keywords**—programming language; visualization; evaluation

### I. 前言

由於程式語言具抽象且複雜的特性，學習程式語言對初學者而言是一門十分困難的學科。因此，許多研究 (Brusilovsky & Spring, 2004; Moreno, Myller & Bednarik, 2005; Shanmugasundaram, Juell, Groesbeck & Makosky, 2006) 嘗試運用視覺化程式開發環境來提供學習者更具體的操作經驗和動態情境介面，使學習能更具有樂趣與成效。視覺化工具的互動及將抽象概念轉化為視覺化動態過程的特性，可以幫助學習者「看到」原本只能憑空想像的抽象過程，藉此來幫助學習者理解抽象、複雜的概念，使學習活動更具有樂趣與成就。先前研究 (Moreno et al., 2005) 也指出，教學者應熟悉視覺化工具的所有可能運用，讓學習者在不同情境下透過觀察視覺化內容並聆聽詮釋，以期能有效幫助初學者對程式語言學習的理解。許多研究 (Ben-Bassat Levy, Ben-Ari & Uronen, 2003, 2007) 更明確指出，

視覺化工具是有效的教學工具，但是必須是整合在課堂練習與指定作業之中，而不只是用來作為教學展示而已。因此，本研究期盼透過運用視覺化工具融入程式設計教學活動，以培養學習者有效運用輔助工具增進對程式語言學習的理解、提升學習樂趣、並促進學習者正向的學習態度。

### II. 文獻探討

#### A. 視覺化程式設計工具

Price、Baecker 與 Small (1993) 認為視覺化工具是包含所有使用視覺化功能，圖形、動畫或是其他科技等，可以幫助或是教導軟體工程(設計與除錯)的程序之工具。視覺化工具設計是針對人基互動，目的是可以讓學習者更容易理解、學習與更有效地使用電腦軟體。Myller (2004) 延伸 Price 等人 (1993) 的研究，將視覺化工具(軟體視覺化)分為兩個次領域：演算法視覺化(algorithm visualization)與程式視覺化(program visualization)。演算法(algorithm)是程式中的抽象化概念或是程式的部份，因此演算法視覺化(或動畫)是以視覺化方式呈現軟體一部分的高階特性 (Price et al., 1993)，這類視覺化的內容通常與任何程式的來源碼(source code)都無關。程式視覺化是著重程式的實際內容，像是程式碼與資料，因此程式視覺化必須定義程式圖像化展現的對應與轉換。Java 程式視覺化工具，研究者將其定義為兩種，第一種是用來呈現程式之靜態結構，例如 *BlueJ*，用來呈現程式之靜態結構之視覺化工具，是針對初學者所設計，也提供教學者在教授程式語言物件導向概念時的一個輔助工具 (Kolling, Quig, Patterson & Rosenberg, 2003)。第二種則是用視覺化方式呈現程式執行時動態之歷程與變化，例如 *Jeliot 3*，是使用 Java 程式語言開發，以動畫方

式呈現程式運作，並使用動畫，以流暢地移動圖形物件來視覺化資料結構與程式執行歷程的一個免費工具。此工具提供自動化動畫顯示功能，也就是說使用者輸入一般的程式碼，*Jeliot 3* 會自動產生並播放動畫(Lattu, Tarhio & Meisalo, 2000)。

### B. 視覺化工具之程式設計教學應用

由於初學者在學習程式設計時，常遇到困難的主要原因有二，一是學習程式設計時需要對抽象概念的知識有正確的理解；二是初學者往往缺乏足夠的資源和適性的指導(Lahtinen, Ala-Mutka, & Järvinen, 2005)。視覺化工具提供學習者更具象化的介面，將有利於程式設計初學者掌握程式設計的邏輯和架構(Myller, 2004)。因此，如何選擇適當的視覺化工具和教學設計以促進學習者的學習成效和學習態度是值得探討與研究的。

Mayer 及 Sims (1994) 一連串多媒體學習實驗中，有一致性的發現，視覺化內容或是工具需要搭配講解才會更有成效，而多媒體的呈現扮演了兩個主要的角色，一個是吸引學習者的注意，另一個則是幫助學習者將文字內容與所學習之概念建立強大的連結。Kehoe、Stasko 與 Taylor (2001) 的研究是使用動畫式的視覺化呈現，並融入課堂教學與回家作業中，而不僅僅是考試而已。其研究結果指出，動畫式的視覺化呈現其教學價值在設計開放式回家作業，較傳統考試有顯著效果。Lattu 等人 (2000) 在多個程式語言課程中，透過觀察、錄影、面談與問卷進行資料收集，以探討哪些教學情境適合運用 *Jeliot*。Ben-Bassat Levy 等人(2003) 也運用類似方法檢視在高中程式設計導論課程中使用 *Jeliot* 的效益，研究發現中等程度之學習者進步較顯著，並且可以用更多專有名詞去描述程式。Myller 與 Nuutinen (2006) 則是評估使用 *Jeliot* 對於合作學習的影響，研究結果顯示僅觀看視覺化是不夠的，還要在視覺化呈現過程中提供有意義的對話、合作與互動的機會。

由於過去運用 *Jeliot* 幫助初學者學習程式設計的研究仍缺少嚴謹的實證探討，因此本研究採用 *Eclipse* 開發環境作為城市預言學習的基本模式，再搭配 *Jeliot 3* 視覺化工具，以檢視視覺化工具對初學者程式設計學習的學習成效與學習態度的影響。

## III. 研究方法

### A. 研究設計

本研究採用準實驗研究法。自變項包含「視覺化工具融入教學」與「先備知識程度」。「視覺化工具融入教學」依據有無採用視覺化工具分為兩組，分別為「*Jeliot* 組」和「*Eclipse* 組」。兩組皆均採用 *Eclipse* 開發環境為基本開發工具，*Jeliot* 組使用 *Eclipse* 開發環境搭配 *Jeliot 3* 視覺化工具融入教學，*Eclipse* 組則僅使用 *Eclipse* 開發環境。「先備知識」為學習者在學習 *for* 迴圈程式設計概念之前應先具備的程式語言知識。在進行學習活動前，先進行先備知識測驗，依據測驗所得之分數，依照分數高低排列分為兩組，前 50% 為高先備知識組，後 50% 為低先備知識組。依變項有二，分別為「學習成效」及「學習態

度」。「學習成效」主要探討學習者在實驗教學活動之後在 *for* 迴圈的學習的表現情形，分為基本概念與進階概念二個面向；「學習態度」主要探討學習者在實驗教學活動之後對於所使用教學活動的看法與感受，包含學習滿意度、學習幫助度、自我效能、學習焦慮和主動學習與思考等面向。

### B. 研究對象

本研究的研究對象為程式語言初學者，研究樣本為大學資管系一年級學生，在實驗進行前皆已學過 *Java* 程式語言基礎語法與 *if* 判斷式之程式語言單元。為配合原班級課程安排，以班級為單位，隨機抽取兩個班級為研究樣本，總共 104 位研究參與者，男生 51 位，女生 53 位，年齡分布於 16~20 歲之間，再以班級為單位隨機將其中兩個班級分為 *Jeliot* 組，共 51 人，另一個班級分為傳統教學組，共 53 人。為求實驗的準確性，剔除 25 位未能全程參與的學習者，因此本研究之有效樣本總共 79 人，男生 42 位，女生 37 位，其中 *Jeliot* 組 43 人和 *Eclipse* 組 36 人。

### C. *Eclipse* 開發環境與 *Jeliot 3* 視覺化工具

本研究在 *Java* 程式語言學習上，控制組(*Eclipse* 組)採用 *Eclipse* 開發環境，實驗組(*Jeliot* 組)則使用 *Eclipse* 開發環境搭配 *Jeliot 3* 視覺化工具。

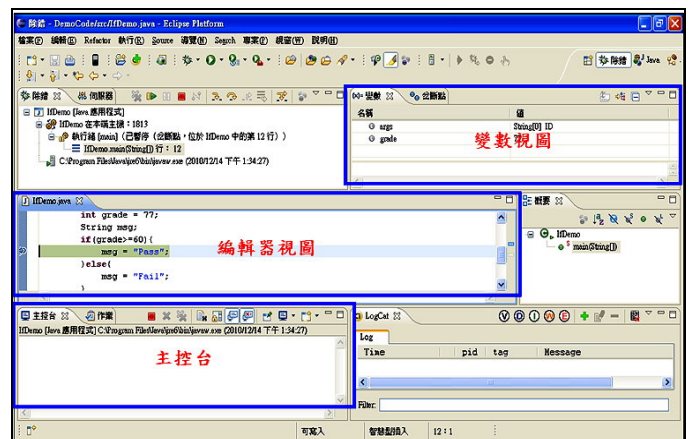


圖 1: *Eclipse* (Ganymede 3.4.2) 開發環境介面

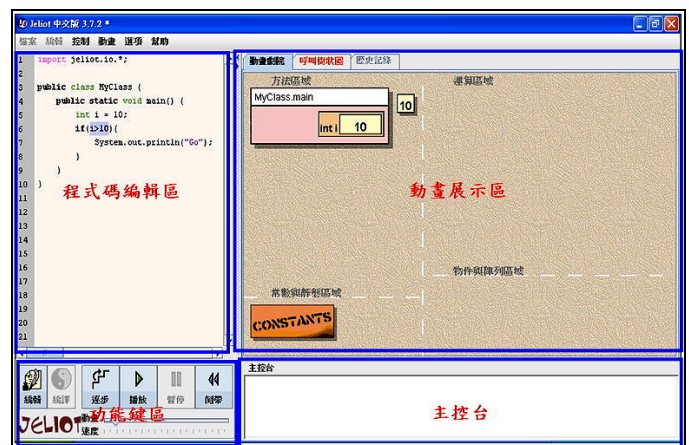


圖 2: *Jeliot 3* 視覺化工具介面

如圖 1 所示, *Eclipse* 開發環境提供有中文化支援。*Eclipse* 開發環境是 IBM *Eclipse* 專案中所提供免費的程式語言開發工具, 提供 Java 或是 C、C++ 程式語言多種版本。本研究採用的版本是可以執行在 Windows 作業系統下的 Java 程式語言版本, 版本別為 Ganymede 3.4.2。如圖 2 所示, *Jeliot 3* 視覺化工具軟體為 Java 程式視覺化工具的一種, 由芬蘭 Joensuu 大學運用 Java 程式語言所開發完成, 是以動畫方式呈現程式運作, 並使用流暢移動圖形物件來視覺化資料結構與程式流程的一個免費工具。由於 *Jeliot 3* 僅有英文版本, 研究者考慮受試者操作度與語言熟悉度, 獲得 *Jeliot 3* 軟體研發團隊同意由研究者自行修改, 開發完成 *Jeliot 3* 多國語言支援版本, 以應用於本研究之實驗教學活動中。

#### D. 研究工具

本研究的研究工具包含：先備知識測驗、成就測驗及學習態度問卷。各工具分別說明如下：

**先備知識測驗**由研究者自行設計, 採用專家效度, 經由兩位相關領域專家審核, 確認該題目符合測試主題, 審閱後修訂完成。先備知識測驗之知識內容為變數命名規則、型別與表示法、運算子與題意理解以及 if 判斷式, 題型為複選題、單選題以及填充題, 總分 21 分。測驗信度經內部一致性考驗, Cronbach's  $\alpha = .712$ , 內部一致性合乎理想。

**成就測驗**乃是以本研究實驗教學內容為範圍, 測驗向度分為迴圈基本語法、迴圈累進值運算、判斷式、運算子與題意理解, 採用專家效度, 經由兩位相關領域專家審核, 確認該題目符合測試主題, 審閱後修訂完成。題型為簡答題、填充題以及問答題, 總分 54 分。測驗信度 Cronbach's  $\alpha = .846$ , 內部一致性合乎理想;

**學習態度問卷**於教學實驗後實施, 目的在於瞭解學生在學完教學實驗課程之後的看法和情形。量表內容分為五個向度：「學習滿意度」、「學習幫助度」、「自我效能」、及「主動學習與思考」, 共 23 題, 採用 Likert Scale 的五點量尺, 每題 1 至 5 分, 各面向分數越高表示學習態度越正面。信度經內部一致性考驗, Cronbach's  $\alpha = .865$ , 內部一致性合乎理想。

### IV. 研究結果

本研究分別就「學習成效」及「學習態度」進行分析, 「學習成效」主要探討學習者在實驗教學後學習表現, 分為基本概念與進階概念二個面向; 「學習態度」則在檢視學習者對於實驗教學活動之學習滿意度、學習幫助度、自我效能、學習焦慮及主動學習與思考等面向。

#### A. 學習成效分析

本研究「視覺化工具」與「先備知識」各組學習者在教學活動後之成就測驗表現如表 1 所示, *Jeliot* 組在基本概念與進階概念表現之平均數皆略高於 *Eclipse* 組; 高先備知識組在基本概念與進階概念表現之平均數也都略高於低先備知識組。

各組成績經多變量變異數分析(MANOVA), 結果顯示基本概念與進階概念之「視覺化工具 X 先備知識」交互作用未達顯著, 基本概念之「視覺化工具」與「先備知識」主要效果亦未達顯著水準, 但「視覺化工具」與「先備知識」之進階概念主要效果則達顯著水準( $F_{(1,74)} = 3.129, p = .045$ ), ( $F_{(1,74)} = 4.280, p = .036$ )。此結果顯示, 就基本概念而言, 學習者不論是使用不同的「視覺化工具」或「先備知識」高低, 都有相同程度的學習表現; 但在進階概念上, 使用 *Eclipse*+*Jeliot* 視覺化工具優於只使用 *Eclipse* 工具、高先備知識學習者也比低先備知識學習者有較佳的表現。

表 1: 各組成就測驗成績之平均數、標準差及人數表

依變項	自變項	平均數	標準差	人數
基本概念 (28 分)	<i>Jeliot</i>	15.459	3.678	43
	<i>Eclipse</i>	13.470	2.373	36
	高先備知識	15.431	3.081	40
	低先備知識	13.652	3.285	39
進階概念 (26 分)	<i>Jeliot</i>	10.762	7.094	43
	<i>Eclipse</i>	7.530	3.771	36
	高先備知識	11.169	6.294	40
	低先備知識	7.361	5.089	39

#### B. 學習進展之分析

本研究進一步比較學習者在前測及後測之學習進展幅度, 先將先備知識測驗及成就測驗成績轉換為總分 100 分之相對分數; 然後以成就測驗成績減去先備知識測驗, 以作為學習進展幅度分數, 並將進展幅度區分為：「退步 5 分以上」、「退步或進步 5 分以內」及「進步 5 分以上」等, 退步、持平及進步三個等級。「視覺化工具」與「先備知識」各組學習者學習進展幅度之比較如表 2 所示, *Jeliot* 組比 *Eclipse* 組有較佳的學習進展, 但是低先備知識學習者則比高先備知識學習者有較大的學習進展幅度。

表 2: 學習進步幅度比例摘要表

自變項	成長分類	所佔比例
視覺化工具	退步(-5~)	<i>Jeliot</i> (26 %) < <i>Eclipse</i> (33 %)
	持平(+5)	<i>Jeliot</i> (23 %) < <i>Eclipse</i> (47 %)
	進步(+5~)	<i>Jeliot</i> (51 %) > <i>Eclipse</i> (20 %)
先備知識	退步(-5~)	Hi-PK (32.5 %) > Lo-PK (27 %)
	持平(+5)	Hi-PK (35.0 %) > Lo-PK (33 %)
	進步(+5~)	Hi-PK (32.5 %) < Lo-PK (40 %)

#### C. 學習態度分析

本研究學習者對所安排教學活動之看法與感受, 分別從「學習滿意度」、「學習幫助度」、「自我效能」、及「主動學習與思考」四個向度進行分析。「視覺化工具」與「先備知識」各組學習者各組學習態度的平均數、標準差及人數如表 3 所示, *Jeliot* 組比 *Eclipse* 組在各面向上都有較佳的學習態度, 但是高、低先備知識的學習者則對各學習態度面向有相當接近的看法。

學習態度之 MANOVA 分析結果顯示，在視覺化工具方面，*Jeliot* 組在學習滿意度( $F_{(1,75)} = 14.477, p < .001$ )、學習幫助度( $F_{(1,75)} = 8.344, p = .005$ )及主動學習與思考( $F_{(1,75)} = 5.534, p = .021$ )等面向的看法上，顯著優於 *Eclipse* 組；就先備知識而言，各面向則未達顯著水準。此結果顯示，採用 *Jeliot 3* 視覺化工具有助於學習，學習者的學習滿意度、學習幫助度和主動學習與思考方面能獲得提升。

表 3: 各組學習態度之平均數、標準差及人數

依變項	自變項	平均數	標準差	人數
學習滿意	<i>Jeliot</i>	3.283	.3701	43
	<i>Eclipse</i>	2.931	.4461	36
	高先備知識	3.108	.4457	40
	低先備知識	3.137	.4408	39
學習幫助	<i>Jeliot</i>	3.423	.3810	43
	<i>Eclipse</i>	3.128	.5262	36
	高先備知識	3.335	.4538	40
	低先備知識	3.241	.4946	39
自我效能	<i>Jeliot</i>	3.014	.5281	43
	<i>Eclipse</i>	2.806	.5903	36
	高先備知識	2.945	.5043	40
	低先備知識	2.892	.6238	39
主動思考	<i>Jeliot</i>	3.279	.4174	43
	<i>Eclipse</i>	3.037	.5056	36
	高先備知識	3.167	.3849	40
	低先備知識	3.171	.5508	39

## V. 結論

本研究發現，「視覺化工具」及「先備知識」對於 Java 程式語言的進階概念學習上具有提升學習成效之效益。首先，就先備知識來看：由於高先備知識學習者相較於低先備知識學習者，其在長期記憶中有更多且具有組織性的基模，在學習新知識時，不論是連結舊經驗或是提取與編碼都更為流暢，因此在進階概念的學習成效上相較於低先備知識學習者可以達到較佳的效益。然而，就學習進展幅度而言，低先備知識學習者由於可以進步的空間比較大，在接受到適切規劃設計的教學活動後也可能達到很好的學習成效，因此就是硬此較學活動的低先備知識學習者而言，其所達到的進展幅度優於高先備知識學習者。再者，就使用 *Jeliot 3* 視覺化學習工具而言，學習者在 Java 程式語言進階概念學習成效上，不但比只使用 *Eclipse* 學習環境的學習者有較佳的學習表現，而且在學習進展的幅度上也比較大，顯示使用視覺化學習工具可以有效促進初學者對於程式內部運作的理解，幫助其建立較精確的基模，進而能有較佳的學習表現。

由學習態度分析結果發現：視覺化工具組學習者在學習過程中遭遇到自己所理解觀念與實際執行結果有明顯差異時，可以藉由 *Jeliot 3* 視覺化工具去檢視程式語言的背後運作歷程，以改正迷思概念，進而獲得學習上的助益。因此，使用 *Jeliot 3* 視覺化工具之學習者可以藉由視覺化工具提供正確且淺顯易懂化的視覺呈現，可以藉由工具所呈現的程式運作與自己內在的理解相互對照與印證，並做

適度的修正，所以在學習滿意度、學習幫助度和主動學習與思考等方面都表現出更為正向的看法。

綜合上述，在程式語言教學活動中適切運用視覺化工具，幫助初學者正確檢視程式背後的運作概念，能夠有效地提升學習者學習程式語言的成效、促進初學者達到較佳的學習進展、並且提升學習的樂趣與正向的學習態度。

## ACKNOWLEDGEMENT

This study is partially supported by the National Science Council, Taiwan, under Grant No. NSC 98-2511-S-003-034-MY3 and the International Collaboration on E-Learning Project (NSC 98-2631-S-110 -001).

## REFERENCES

- [1] Ben-Bassat Levy, R. & Ben-Ari, M. (2007). We work so hard and they don't use it: Acceptance of software tools by teachers, *ACM SIGCSE Bulletin*, 39(3), 246-250.
- [2] Ben-Bassat Levy, R., Ben-Ari, M. & Uronen, P. A. (2003). The Jeliot 2000 program animation system. *Computers & Education*, 40(1), 15-21.
- [3] Brusilovsky, P. & Spring, M. (2004). Adaptive, engaging, and explanatory visualization in a C programming course. *Proceedings of the 2004 World Conference on Educational Media, Hypermedia, and Telecommunications*, 1264-1271.
- [4] Kehoe, C. M., Stasko, J. T. & Talor, A. (2001). Rethinking the evaluation of algorithm animations as learning aids: An observational study, *International Journal of Human Computer Studies*, 54, 265-284.
- [5] Kolling, M., Quig, B., Patterson, A. & Rosenberg, J. (2003). The BlueJ system and its pedagogy. *Journal of Computer Science Education*, 13(4), 249-268.
- [6] Lahtinen, E., Ala-Mutka, K. & Järvinen, H. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, 37(3), 14-18.
- [7] Lattu, M., Tarhio, J. & Meisalo, V. (2000). How a visualization tool can be used: Evaluating a tool in a research and development project. In *12th Workshop of the Psychology of Programming Interest Group* (pp.19-32), Corenza, Italy.
- [8] Mayer, R. E. & Sims, V. K. (1994). For whom is a picture worth a thousand words? Extensions of a dual-coding theory of multimedia learning. *Journal of Educational Psychology*, 86(3), 389-401.
- [9] Moreno, A., Myller, N. & Bednarik, R. (2005). Jeliot 3, an extensible tool for program visualization. *5th Annual Finnish / Baltic Sea Conference on Computer Science Education*.
- [10] Moreno, A., Sutinen, E., Bednarik R. & Myller, N. (2007). Conflictive animations as engaging learning tools. In R. Lister and Simon, editors, *Seventh Baltic Sea Conference on Computing Education Research (Koli Calling 2007)*, 88, 203-206.
- [11] Myller, N. (2004). *The fundamental design issues of Jeliot 3*. Unpublished master's thesis, University of Joensuu, Department of Computer Science.
- [12] Myller, N. & Nuutinen, J. (2006). JeCo: Combining program visualization and story weaving. *Informatics in Education*, 5(2), 195-206.
- [13] Price, B. A., Baecker, R. M. & Small, I. S., (1993). A principled taxonomy of software visualization. *Journal of Visual Languages & Computing*, 4(3), 211-266.
- [14] Shanmugasundaram, V., Juell, P., Groesbeck, G. & Makosky, M. (2006). Evaluation of Alice world as an introductory programming language. *Proceedings of the ED-MEDIA 2006-World Conference on Educational Multimedia, Hypermedia & Telecommunications*, 1976-1982.