

## 設計遊戲程式對培養小學生問題解決能力的效果

### The Effect of Digital Game Programming on the Problem-Solving Skills of Elementary School

#### Students

黃筱琪，邱瓊慧\*

國立臺灣師範大學資訊教育研究所

\*cchui@ntnu.edu.tw

**【摘要】** 本研究的目的是探討設計遊戲程式對培養小學生問題解決能力的效果。本研究邀請北臺灣某國小共 93 位五年級學生參與「設計遊戲程式課程」，活動中學生需要共同使用一多點觸控平台完成設計遊戲程式專題。為瞭解學生問題解決能力提升的情形，本研究在活動前、後請學生填寫問題解決能力量表，前、後資料經成對樣本 t 檢定後，顯示參與活動學生的問題解決能力有顯著進步，亦即「設計遊戲程式課程」能提升學生的問題解決能力 ( $p < .05$ )。不過，因本研究採單組前後測，可能有前測經驗影響的問題，且無法確定前、後測之間是否有其他事件影響實驗結果。

**【關鍵字】** 遊戲程式；問題解決能力；小學科學教育

*Abstract: The purpose of this study was to investigate the effect of digital game programming on the problem-solving skills of elementary school students. Ninety-three fifth-grade students from an elementary school in northern Taiwan participated in this study. The students worked in groups of three or four to complete a digital game programming project supported by a multi-touch screen desktop computer. A pretest-posttest design was adopted to investigate the level of problem-solving skills. The results of paired sample t-test showed that digital game programming could improve the problem-solving skills of elementary school students. However, it's difficult to determine whether any observed changes from pretest to posttest were due to the treatment or whether they were the result of other extraneous variables.*

**Keywords:** game programming, problem-solving skills, elementary science education

## 1. 前言

### 1.1 研究背景

在資訊化的社會中，培養國民具備運用資訊科技的基本知識與技能，已然是世界各國教育發展的共同趨勢，利用數位遊戲的特性：動態影像、衝突及挑戰、互動和及時回饋等 (Felix & Johnston, 1993; Baranauskas, Neto & Borges, 2001)，將學習的概念設計到數位遊戲中，藉由適性化的挑戰任務引發學習者主動參與，使學習者從遊戲中學習到概念，進而提升其學習動機及學習成效，達到寓教於樂的學習目標，將教學目標融入遊戲中，可克服教室中教學枯燥乏味的困境。此外，Prensky 認為 21 世紀為數位遊戲學習 (digital game-based learning) 的世代，學生接觸數位遊戲的機會大幅增加，數位遊戲不再只是扮演娛樂的功能，其所帶來的影響性開始受到廣泛的討論與研究，學者 Rosas 等人整理過去數位遊戲式學習的相關研究，指出數位遊戲式學習對於學習成就、認知能力發展、學習動機、及學習專注力等四大方面皆有所助益 (Rosas et al., 2003)。

透過設計遊戲來進行各種學科的學習，是各國教育研究者正在嘗試的方向 (孫春在, 2013)，從學習的角度來看，設計是一種高層次的概念整合，在學習過程中有助於激發「高層次思考」

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.

(high-order thinking) (孫春在, 2013), 也就是批判思考、創造思考、後設認知, 以及學習技巧。對於學習程式語言的目的, MIT Media Lab 計劃執行長 Mitch (2012) 認為重點並非是要將所有年輕人培訓成程式設計師, 而是透過學習程式語言, 開拓更寬廣的學習途徑, 「當你學會閱讀, 你便能藉著閱讀學習更多知識, 程式設計也是一樣的道理; 如果你會撰寫程式, 你能透過程式語言學習到的事物將更為多樣。」; Apple 創辦人 Steve Jobs (2013) 亦指出「這個國家的每一個人都應該要學電腦程式, 因為這會教你怎麼思考」, 學寫程式就是在學習創意思考、有系統的推論、和團隊合作, 而這些技能不僅在各專業領域都受用無窮, 更是生活中不可或缺的能力。

從以上種種國際趨勢顯示, 在不久的未來, 懂程式語言就跟會算數、閱讀一樣, 將是人人必備的「基本能力」, 學習程式設計能夠培養學生問題解決以及高層次思考能力 (Georgios, Evangelia & Elisavet, 2013), 程式設計不只是一堆指令的組合, 它代表一連串的問題解決步驟 (Winslow, 1996), Funkhouser (1993) 也指出在學習程式設計的過程中, 如果能提供適當的程式設計工具幫助學習者將問題視覺化或物件化, 將能促進學習者的思考與解題。根據經濟合作暨發展組織所公布的「2015 年 PISA 合作解決問題的評量框架」, 國際學生能力評量計畫 (the Programme for International Student Assessment, 簡稱 PISA) 的評量方式除了全面電腦化測驗, 更首度加測學生的線上「合作問題解決能力」, 包含三種基本能力: 一、分享資訊、達成共識; 二、解決問題的方式 (主動提問、確認); 三、建立與維持團隊運作 (確認進度、分工), 由此可知, 問題解決能力為二十一世紀人才的重要能力之一。為此本研究旨在融入設計遊戲程式課程於國小階段, 希望透過視覺化的程式設計工具培養學生的問題解決能力, 並根據研究結果提出具體可行的建議, 提供有意導入相關課程的教育工作人員或是有意願從事或設計數位遊戲式學習之相關教學者與研究者做為參考。

## 1.2. 研究目的

本研究旨在探討設計遊戲程式對培養小學生問題解決能力的效果, 欲探討的問題為: 設計遊戲程式對小學生的問題解決能力是否能造成顯著的改變?

## 2. 研究方法

本研究以單組前後測實驗設計檢視並比較小學生在參與設計遊戲程式課程前後, 其問題解決能力的變化情形。

### 2.1. 參與者

本研究的參與者為台北市中正區某國小五年級四班共 93 位的學生 (男 53 人, 女 40 人), 本研究配合學校原定的課程安排, 徵得校方及電腦科教師的同意, 由研究者 (本文作者之一) 以原班級為單位進行教學。參與者於三、四年級均學習過基本電腦的使用, 如鍵盤滑鼠操作、基本視窗使用、及文書處理軟體等, 但未曾在學校學過「程式設計」相關內容。

### 2.2. 研究工具與資料收集

#### 2.2.1. 遊戲設計軟體-Kodu

Kodu 為微軟專為學童設計的視覺化程式語言 (visual programming language) 與遊戲開發工具, 透過設計遊戲的任務方式幫助學生學習程式設計。Kodu 以直覺式的圖示進行操作 (如圖 1), 以及修改角色、物件的屬性激發學童設計遊戲的興趣及成就感, 使學習者即使沒有程式設計的先備知識也能創造屬於自己的 3D 遊戲 (如圖 2)。

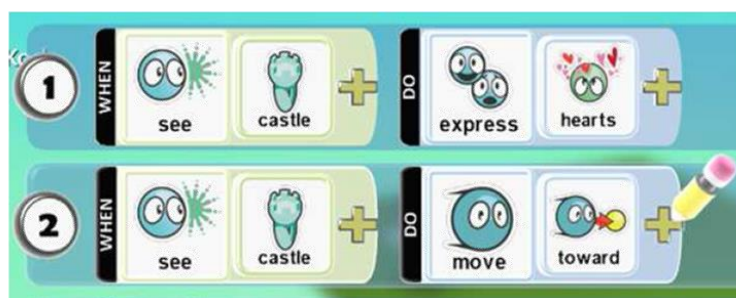


圖 1 Kodu 直覺式操作介面



圖 2 學生共同操作 Kodu 平台，設計遊戲程式

### 2.2.2. 問題解決能力量表

本研究以問題解決能力量表測量學生在參與設計遊戲程式課程前後問題解決能力的改變。此量表由研究人員所屬研究團隊所研發，乃參考 Fortus 等人（2005）評估學生參與設計式學習活動之問題解決能力的六個面向，及 Heppner 與 Petersen（1982）、潘怡吟與王美芬（2003）的問題解決量表題項所設計而成，量表包含 23 題正向題和 3 題反向題，共 26 題，題目以李克特五點量尺設計，學習者可針對題目陳述的內容表達「非常符合」、「符合」、「少部分符合」、「不符合」、或「非常不符合」等看法，且內容的適切性與完整性經三位資訊教育暨合作學習領域學者檢核確定，完整作答時間需時 15 分鐘，學生可得分範圍為：26 至 130 分。該量表曾用於電腦化合作腳本對小學學生進行多點觸控支援合作設計式學習的投入程度、合作技巧與問題解決能力影響之路徑分析研究（陳政煥、邱瓊慧，2014），其報導的 Cronbach's  $\alpha$  值為.94。本研究前、後測所得資料的 Cronbach's  $\alpha$  值分別為.94、及.93。

### 2.2.3. 非正規訪談

本研究於每節教學活動結束時，對參與研究的學生進行非正規訪談，作為進一步探索設計遊戲程式課程實施效果的依據。本研究隨機選擇每班各三組的學生，於實驗活動結束後進行訪談，以詢問學生對於設計遊戲程式課程的想法。訪談問題如下：

1. 經過這次的活動，我對於設計遊戲程式有什麼想法？
2. 是否希望未來能再參與這種在多點觸控平台上的設計活動？為什麼？
3. 我對於這次的活動有什麼想法與建議呢？

### 2.3. 實驗流程

實驗活動包括實驗前訓練、前測、實驗處理、及後測四個部分，於各班每週一節的電腦課（每節 40 分鐘）進行，由研究者擔任教學者，架設七部中尺寸平台式多點觸控螢幕及電腦主機，支援學生進行設計遊戲程式課程，各週實驗活動內容進一步說明如下：

1. 實驗前訓練。研究者以一堂課時間教導學生操作多點觸控平台及引導學生使用遊戲設計軟體 Kodu 完成一個簡單射擊遊戲的任務，讓學生熟悉 Kodu 介面上的增加物件、設定

程式、以及儲存檔案等功能。教學流程及時間分配為：於教室前方投影教學簡報與注意事項（15 分鐘）；講解合作及設計相關評分規則（10 分鐘）；並引導學生認識 Kodu 介面（15 分鐘）。

2. 前測。前測於實驗處理前一週實施，研究者請學生填寫問題解決能力量表，檢視學生在進行設計遊戲程式課程前的問題解決能力。

3. 實驗處理。研究者安排學生三或四人一組，小組成員共同使用一多點觸控平台進行為期四週的設計遊戲程式課程（如圖 3）。本課程以「Kodu」作為設計遊戲程式的平台，共包含：澄清問題（一週）、蒐集資料（一週）、建構作品（一週）、與發表作品（一週）等四個階段，在「澄清問題」階段，研究者先口頭介紹設計情境以引起學生學習動機，再讓學生閱讀專案設計說明書，接著，在研究者說明專案的評量標準後，學生依據專案設計說明書的要求進行小組討論，並條列出專案設計的問題與需求；在「蒐集資料」階段，學生先個別瀏覽研究者所提供的參考資料，其內容為程式設計的基本概念，蒐集與擷取設計專案時所需要的可能資料，再與組員討論前一階段中專案設計的問題、需求與可能資料之間的關聯，並共同條列出經過思考與討論的整理結果；在「建構作品」階段，學生依據前兩階段的成果，以小組合作的形式在多點觸控平台上建構遊戲程式（如圖 4）；在「發表作品」階段，各組成員上台發表該組的遊戲程式設計內容（如圖 5），接著進行遊戲程式作品交互試玩，最後，由研究者總結整個設計遊戲程式課程，並提供學生整體性的回饋與指導。

4. 後測。後測於實驗處理後一週實施，後測的測驗內容與前測相同，為問題解決能力量表之填寫，以瞭解學生在進行設計遊戲程式課程後的問題解決能力。

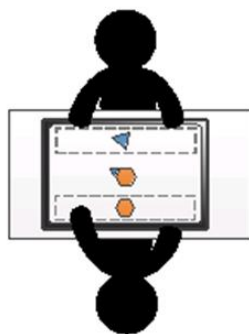


圖 3 學生三或四人一組，共同使用一多點觸控平台



圖 4 學生建構作品情形



圖 5 學生設計遊戲程式的作品畫面

### 3. 研究結果與討論

本研究針對學生「問題解決能力量表」的前、後測資料進行資料的分析與處理，利用成對樣本  $t$  檢定探討學生在參與課程前後問題解決能力的改變情形，以瞭解學生的問題解決能力是否進步，並輔以學生訪談、教室觀察記錄等質性資料加以分析與探究。

表 1 呈現學生參與「設計遊戲程式課程」前、後問題解決能力量表的總分，學生問題解決能力前測平均得分為 95.22，問題解決能力後測平均得分為 99.40。

表 1 問題解決能力之描述行統計摘要表

	平均數	$n$	標準差	標準錯誤平均值
前測	95.22	93	18.40	1.91
後測	99.40	93	14.65	1.52

註： $n = 93$ 。

表 2 呈現「問題解決能力」成對樣本  $t$  檢定結果， $t$  值為 2.689， $p = .009$ ，代表學生問題解決能力得分之前後測差異（4.18 分）達顯著水準（ $p < .05$ ），亦即實施「設計遊戲程式課程」有助於提升學生的問題解決能力。

表 2 問題解決能力之成對樣本  $t$  檢定

	平均數	標準差	標準錯誤平均值	$t$	$df$	顯著性 (雙尾)
前測-後測	-4.18	15.00	1.56	-2.69	92	.009

此外，本研究針對學生訪談資料與教室觀察紀錄進行分析，發現多數學生對於設計遊戲程式的想法，從初期覺得困難，隨著課程進行到中後段時，轉變為樂於接受且勇於嘗試，研究者推論其轉變原因可能為本研究的遊戲程式設計工具 Kodu 為視覺化程式語言工具，學生能透過圖像式操作，組合出所需之程式語言，故學生能對於程式設計保持高度的學習動機與興趣，此與 Funkhouser (1993) 的研究發現一致，若提供適當的程式設計工具，將能促進學習者的思考與學習興趣，尤其是針對從未接觸程式設計的初學者。

另外，本研究追蹤學生對於設計遊戲程式課程的想法之訪談資料（如表 3），發現多數學生（80%）認為設計遊戲程式課程有助於認識程式設計的基礎概念，S01-27 學生表示：他們能依據所學得的概念，設計出專屬的遊戲程式；S01-23 學生也稱：他們可運用以前所學的知

識來解決問題，當研究者進一步詢問，S01-23 學生甚至認為：程式設計概念可與以前所學知識相結合，並運用於實際生活中。

表 3 學生對遊戲程式的想法

學生	回答內容
S01-01	我可以學習很多程式設計的新概念。
S01-23	可以利用以前學的知識來解決問題。
S01-27	我學會新的程式設計概念，也讓我學到了如何設計遊戲程式。
S02-04	可以讓你學會設計遊戲程式。
S02-10	這次的課程好有趣，我學到了各種新的概念。
S03-24	設計遊戲程式十分有趣，還可以動頭腦，我獲益良多。
S03-27	讓我學到程式設計。
S04-04	可以學到一些新知識。
S04-22	我覺得我學到更多有關程式設計的知識。

表 4 呈現學生對於專題式設計遊戲程式課程的想法，可以發現，多數學生（90%）喜歡本專題式課程活動，認為此種上課方式有別於一般傳統上課方式，不只有趣、好玩，更提供較多的機會思考，並且發揮自己的創造力，表 4 中 S04-01 學生更表示設計專題活動讓他發現設計一個遊戲沒有想像中的容易，因為，過程中需經過多次測試，以驗證遊戲程式是否與自己的想法相吻合。此與研究者的期待相符，學生的確在專題課程中實際創造遊戲程式以解決真實世界中的課題，如同程式設計中十分重要且繁瑣的「debug」過程，此過程也正是考驗學生問題解決能力的重要指標。

表 4 學生對設計遊戲程式專題的想法

學生	回答內容
S01-01	不像在一般課堂上，我喜歡這種上課方式。
S01-10	這種課程很有趣。
S02-11	讓我創意變多。
S03-04	很好玩，希望還可以再有這種上課方式。
S03-10	讓我能思考更多。
S04-01	設計沒有我們想的那麼容易，他要經過許多的方法來驗證是否 ok。
S04-12	可以發揮自己的創造力。

#### 4. 結論與建議

本研究發現學生在實施「設計遊戲程式課程」前後能力顯著的提升學生的問題解決能力，因此，可以合理推斷，「設計遊戲程式課程」能有效培養學生問題解決能力。不管如何，由於本研究採單組前測-後測設計，實驗結果可能受到前測效應與同時事件的影響，導致內在效

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.

度降低。故本研究建議，未來的研究能增加一對照組作為比較，以解決單組前測-後測設計內在效度較低的問題。此外，訪談資料顯示，「設計遊戲程式課程」可以輔助學生學習程式設計的相關基礎概念，例：變數、迴圈等觀念，但大多數學生反應學習週數不夠，希望能有更多設計遊戲程式的時間，因此，本研究建議未來的教學應用或可考慮增加「建構作品」階段的學習週數，讓學生有更充裕的時間產出作品。另外，教室觀察記錄顯示，多數組別在活動過程中，會將所蒐集的資料與自身經驗相結合，例如：學生將程式設計的基本概念—循序結構，比擬為早上起床後先刷牙再吃早餐等日常生活中例行公事之先後順序，由此可推測，學生面對自己不了解的字詞（循序結構）時，會利用已知詞彙或概念（先後順序）加以詮釋，這樣的過程即是學生將習得的知識內化與精緻化的歷程，因此，本研究建議設計數位遊戲式學習之相關教學者與研究者，建構課程內容或選擇教學素材時，應考慮學生的生活經驗與曾經學過的舊知識，幫助學生強化新舊知識的結合。

## 致謝

感謝行政院國家科學委員會科學教育發展處對本研究經費的補助，計畫編號為：NSC 101-2511-S-003-033-MY3。

## 參考文獻

- 孫春在 (2013)。《遊戲式數位學習》(第三版)。臺北市：心理。
- 陳政煥、邱瓊慧 (2014)。電腦化合作腳本對小學學生進行多點觸控支援合作設計式學習的投入程度、合作技巧與問題解決能力影響之路徑分析。論文發表於「第十屆台灣數位學習發展研討會」，臺北。
- 潘怡吟、王美芬 (2003)。遊戲型態教學對國小學生「自然與生活科技」學習之研究。《臺北市立師範學院學報：人文藝術類、社會科學類、科學教育類》，34，157-172。
- Baranauskas, C. C., Neto, N. G. G., & Borges, M. A. F. (2001). Learning at work through a multi-user synchronous simulation game. *International Journal of Continuing Engineering Education and Life Long Learning*, 11(3), 251-260.
- Csikszentmihalyi, M. (1975). *Beyond boredom and anxiety*. San Francisco: Jossey-Bass.
- Felix, J. W., & Johnson, R. T. (1993). Learning from video games. *Computers in the Schools*, 9(2-3), 119-134.
- Fortus, D., Krajcik, J., Dersheimer, R. C., Marx, R. W., & Mamlok-Naaman, R. (2005). Designbased science and real-world problem-solving. *International Journal of Science Education*, 27(7), 855-879.
- Fowler, W. A. L., & Fowler, R. H. (1993). A hypertext approach to computer science education unifying programming principles. *Journal of Multimedia and Hypermedia*, 2(4), 433-441.
- Funkhouser, C. P. (1993). The influence of problem solving software on student attitudes about Mathematics. *Journal of Research on Computing in Education*, 25(3), 339-346.
- Georgios, F., Evangelia, G., & Elisavet, M. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63(1), 87-97.
- Heppner, P. P., & Petersen, C. H. (1982). The development and implications of a personal problem-solving inventory. *Journal of Counseling Psychology*, 29(1), 66-75.

- 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.
- Johnson, D. W., & Johnson, R. T.(1974). Instructional goal structure: Cooperative, competitive, or individualistic. *Review of educational research, 44*(2), 213-240.
- March, J. G.(1991). Exploration and exploitation in organizational learning. *Organization Science, 71-87*.
- Mitch, R.(2012). *Let's teach kids to code*. Retrieved November, 2012, from [https://www.ted.com/talks/mitch\\_resnick\\_let\\_s\\_teach\\_kids\\_to\\_code?language=en](https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=en)
- Rosas, R., Nussbaum, M., Cumsille, P., Marianov, V., Correa, M., Flores, P.,... Salinas, M. (2003). Beyond Nintendo: Design and assessment of educational video games for first and second grade students. *Computers & Education, 40*(1), 71-94.
- Steve, J.(2013). *What most schools don't teach*. Retrieved February, 2013, from <https://code.org/educate/inspire>.
- Winslow, L. E.(1996). Programming pedagogy—A psychological overview. *SIGCSE Bulletin, 28*, 17-22.