USING VISUALIZATION AS AN EASY WAY TO LEARN PROGRAMMING LOGIC

LOO SZE PHEI

UNIVERSITI UTARA MALAYSIA

MSc. (IT) 2009

PERMISION TO USE

This thesis presents a partial fulfillment of the requirement for a postgraduate degree from Universiti Utara Malaysia. I agree that the university library may make it freely available for inspection. I further agree that the permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by my supervisor or, in their absence by the Assistant Vice Chancellor of the College of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part should be addressed to:

Assistant Vice Chancellor
College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok
Kedah Darul Aman.

ABSTRAK

Dalam Teknologi Informasi (TI) pendidikan, dari peringkat sijil pendidikan di perguruan tinggi atau peringkat darjah institions sampai di universiti, kursus pengaturcaraan sangat penting untuk membina pelajar berfikir logis dan untuk melatih keterampilan program mereka dalam mencipta pelbagai jenis perisian. Malangnya, kursus pengaturcaraan sering kali terdiri daripada tingkat kegagalan lebih tinggi daripada program-program studi ilmu komputer lainnya. Dalam kajian ini, masalah dalam mempelajari kursus pengaturcaraan telah dianalisis melalui kajian literatur. Untuk meningkatkan dan untuk membantu memahami esensi pengaturcaraan komputer, cara mudah untuk mempelajari logika pemrograman dirumuskan berfokus pada penyediaan visualisasi untuk membantu belajar. Pendekatan ini telah diterapkan pada prototaip Courseware, disajikan dan dievaluasi oleh tahun pertama mahasiswa diploma ilmu komputer. Keputusan evaluasi menunjukkan bahawa sebahagian besar pelajar dapat memahami materi yang disampaikan dan mereka bersetuju bahawa menggunakan prototaip Courseware dengan ciri visualisasi penggabungan membantu mereka untuk mengembangkan kemahiran program mereka secara berkesan.

ABSTRACT

In the Information Technology (IT) education, from certificate level in colleges or educational institutions until degree level in universities, programming courses are essential to build students‘ logical thinking and to train their programming skills in creating different types of software. Unfortunately, programming courses often consist of higher failure rates than other computer science courses. In this research, problem in studying programming courses will be analysed through literature review. To improve and to aid the understanding of the essence of computer programming, an easy way to learn programming logic will be formulated focusing on providing visualization to aid the learning. This approach will be applied on a courseware prototype, presented and later evaluated by first year diploma computer science students. Brief on the results obtained.

Keywords: *programming logic, programming language, visualization, courseware, courseware prototype, questionnaire design, logical thinking, abstract thinking ability.*

ACKNOWLEDGEMENTS

First of all, my most profound thankfulness goes to my final project supervisor Dr Nor Laily binti Hashim for her help, guidance and encouragement. I would also like to thank her continuous faith and support in me. Without her encouragement and guidance, it will not be easy for me to reach this extend in completion my report.

Secondly, I would like to thank all my dearest family members, especially to my parents, brothers who have given me their full support in my study. Their full support remains the mainstay for me in overcoming all the difficulties in completing this study. Next I would like to thank all the lecturers who have taught me before throughout the Masters Degree course because the knowledge they imparted have allowed me to be more knowledgeable and thus in a better position to complete this research.

Lastly, I would like to thank all my friends who had given me emotional support and taken care of me at times of difficulties, especially thanks to Ms. Lim Hooi Szu, Ms Khor Jia Yun, Ms. Kam Su Ning, Mr. Tay Shu Shiang and Mr. Ang Shao Long in advising and guiding me in the process of completing this report.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DIT	Diploma in Internet Technology
IT	Information Technology
RAM	Random Access Memory
SPM	Sijil Pelajaran Malaysia
TARC	Tunku Abdul Rahman College

CHAPTER 1

INTRODUCTION

In computer science context, programming can be defined as creating a sequence of instructions to enable the computer to do something [1]. In IT education, programming courses always been thought as the most difficult courses to study. Many students could not grip on the programming logic in the process of constructing a computer program. Programming logic is a method of reasoning in computing program to perform complex computer functions. Thus, especially for first year computer science students, they will find it is hard to study.

Nevertheless, programming is very important skill in computer science field. The basic goal of the programming course is to provide students with a general body of knowledge about programming languages and the ability to reason critically about the application of programming languages in software development. This course also serves as a foundation for further study in areas such as programming language design and language processor implementation [2].

1.1 Problem Statement

As stated in [9], most of the students who start to learn the principles of programming are taught by the old classic approach, which is based on using a general-purpose programming language such as Pascal, Modula-2, LISP, or C, a professional programming environment for the chosen language, and a set of problems from the area of number and symbol processing. The authors found that using general-purpose languages creates some obstacles for novice programmers, where these languages are too big and too particular in its concepts and programming principles. Besides that, they are oriented on number and symbols processing. Moreover, these general-purpose languages provide little leverage for understanding

The contents of the thesis is for internal user only

REFERENCES

[1] WordNet 3.0 (2008). [Online]. Available: http://www.thefreedictionary.com/programming

[2] Bruce, C., Buckingham, L., Hynd, J., McMahon, C., Roggenkamp, M. & Stoodley, I. (2004). *Ways of Experiencing the Act of Learning to Program: A Phenomenographic Study of Introductory Programming Students at University*. Journal of Information Technology Education, Volume 3, pp. 143 – 160.

[3] Eckerdal, A. & Berglund, A. (2004). *What Does It Take to Learn 'Programming Thinking'?*. Journal of Information Technology Education Volume 3, ACM**.**

[4] Feronato, E. (2006). *10 tips that will help you when you are about to learn a new language* [Online]. Available: http://www.emanueleferonato.com

[5] Gal-Ezer, J. (1998). *Teaching Software Designing Skills*. The Open University of Israel.

[6] Nakashima, T., Matsuyama, C., & Ishii, N. (2007). *Analysis of Source Codes Created by Beginners in Programming Education*. Eighth ACIS International Conference on Volume: 2 July 30 2007 - Aug. 1 2007, pp. 774-78.

[7] Shindo, Y. (2000). *Programming education based on computer graphics animation*. Proceedings. International Workshop on 2000, pp. 292-293.

[8] Matsuda, H. & Shindo, Y. (2001). *Effect of using computer graphics animation in programming education*. Proceedings. IEEE International Conference on 2001, pp. 164-165.

[9] Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., & Miller, P. (1997). *Mini-languages: A Way to Learn Programming Principles*. Education and Information Technologies 2 (1), pp. 65-83.

[10] Leo, N. (2003). *How hard is it to learn programming?* [Online]. Available: http://ask-leo.com

[11] Admin of Geek Files (2007). *5 Tips on Improving Programming Logic* [Online]. Available: http://www.sunilb.com/programming/5-tips-on-improving-programming-logic

[12] Ohki, M. & Hosaka, Y. (2003). *A program visualization tool for program comprehension*. Proceedings. 2003 IEEE Symposium on 28-31 Oct. 2003, pp. 263- 265.

[13] Malhotra, N. K. (2006). *Chapter 5 - Questionnaire Design And Scale Development* [Online]. Available: http://www.terry.uga.edu/~rgrover/chapter_5.pdf

[14] TARC Unit & Instructor Evaluation Survey form

[15] Whitney, D. R. (1972). *The Questionnaire as a Data Source* [Online]. Available: http://wbarratt.indstate.edu/documents/questionnaire.htm

[16] Shelly, G. B., Cashman, T. J., & Vermaat, M. E. (2008). *Microsoft Office 2007: Introductory Concepts and Techniques*. Thomson Course Technology.

[17] O'Leary, T. J., O'Leary & L. I., Lee K. M. (2007). *Microsoft Excel 2007*. McGraw Hill.

[18] Lowe, D. (2007). *Microsoft Office Power Point 2007 for Dummies*. Willey Publishing, Inc.

[19] Satzinger, J., Jackson, R. & Burd, S. (2009). *System Analysis & Design In A Changing World 5th edition*. Course Technology.

[20] Yasumura, M., Arisawa, M. & Saito, N. (1991). *A Case Study of Computer Literacy Education*. IPSJ, Vol.32, No.12. Japan, Dec. 1991, pp. 1310-1316.

[21] Matsuyama, C., Nakashima, T. & Ishii, N. (2004). *An Attempt of Making Program-Generated Animation in a Beginners' Programming Class*. IEEJ Trans. EIS, Vol.124, No.12. Japan, Dec. 2004, pp. 2482-2488.

[22] Matsuyama, C., Nakashima, T. & Ishii, N. (2005). *Animation Creation in Computer Software Programming Education and its Evaluation*. Proceedings 6th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel /Distributed Computing (SNPD), 2005, pp. 346-351.

[23] Brusilovsky, P., Grady, J., Spring, M. & Lee, C. H. (2006). *What Should Be Visualized?*. The SIGCSE Bulletin 44 Volume 38, Number 2, 2006.

[24] Naps, T.L., Eagan, J.R., & Norton, L.L. (2000). *An environment to actively engage students in Web-based algorithm visualizations*. ACM SIGCSE bulletin. 32, 1 (2000), pp. 109-113.

[25] Kumar, A.N. (2003). *Model-based generation of demand feedback in a programming tutor*. In: Kay, J. (ed.) Supplementary Proceedings of the 11th International Conference on Artificial Intelligence in Education (AI-ED 2003). IOS Press, Amsterdam, 2003, pp. 425-432.

[26] Brusilovsky, P. and Su, H.-D. (2002). *Adaptive Visualization Component of a Distributed Web-based Adaptive Educational System*. In: Intelligent Tutoring Systems. Vol. 2363. Springer-Verlag, Berlin, 2002, pp. 229-238.

[27] Ma, L., Ferguson, J., Roper, M., Ross, I., & Wood, M. (2008). *Using Cognitive Conflict and Visualisation to Improve Mental Models Held by Novice Programmers*. SIGCSE'08, March 12–15, 2008.

[28] (2009, Sept.) *Educational software* [Online]. Available: http://en.wikipedia.org/wiki/Courseware#Courseware

[29] Marshall, A. D. (1994 - 2005). *Programming in C UNIX System Calls and Subroutines using C* [Online]. Available: http://www.cs.cf.ac.uk/Dave/C/CE.html

[30] (1995, Jan.) *C programming courseware Web site* [Online]. Available: http://www2.its.strath.ac.uk/courses/c/

[31] Brown, B. (1984-2000). *C Programming, v2.7*. [Online]. Available: http://physinfo.ulb.ac.be/cit_courseware/cprogram/default.htm