Examining the role of cognitive load when learning to program.

Keith Nolan
Department of Computer Science
Eolas Building
Maynooth University
keith.nolan@nuim.ie

Aidan Mooney*
Department of Computer Science
Eolas Building
Maynooth University
aidan.mooney@nuim.ie

Susan Bergin*
Department of Computer Science
Eolas Building
Maynooth University
susan.bergin@nuim.ie

ABSTRACT

Cognitive load is concerned with the amount of mental effort imposed on working memory at an instant of time. Changes in cognitive load cause very small dilations of the pupils. The aim of this paper is to examine the role of cognitive load while learning to program through the use of remote eye tracking. Although numerous studies have been carried out to evaluate cognitive load using this approach, very few can be found that have focused on programming comprehension especially with students learning to program for the first time. This study will develop a suite of programming tasks, designed to induce different levels of cognitive load (low to high). The programming tasks will be completed by novice programmers whilst a remote eye tracking system monitors pupil dilation. It is hypothesised that, once environmental factors (ambient light etc) have been controlled, programming tasks designed to induce a high level of cognitive load will result in dilations of the pupils, whilst easier tasks will not result in such a change.

Keywords

Eye tracking, cognitive load, pupillometry, programming comprehension

1. INTRODUCTION

Cognitive science is concerned with how processes of the mind work including the mental processes behind learning, memory and problem-solving [1]. Cognitive load is the total amount of mental activity (processing effort) imposed on working memory at any given time.
Working memory is one of three types of memory that humans have. Long term memory holds a permanent and large body of knowledge and skill. This would include many everyday things that we take for granted such as knowing how to walk or ride a bicycle, how to perform mathematical operations like addition and subtraction, and even recalling simple things such as where we live [1]. Sensory memory is a short term memory which acts as a buffer for the stimuli received through the five senses. However, of most importance to cognitive load is working memory. Working memory provides an interface between long term memory and the senses [1]. It is the section of memory responsible for our creative and logical thinking and allows us to solve problems including programming comprehension. The goal of this study is to investigate the implications of cognitive load on learning to program.

1.1 Measuring cognitive load

Cognitive load can be measured by performance on primary and secondary cognitive tasks with targeted questionnaires. For example, the NASA-TLX survey attempts to gauge the participants perceived cognitive load [2].Physiological measurements can also be used to determine cognitive load with eye tracking showing considerable value as an inexpensive and effective method of measurement. Of particular importance is the relationship between pupil dilation and cognitive load. Changes in cognitive load cause very small dilations of the pupils and in controlled settings, high-precision pupil measurements can be used to detect small differences in cognitive load at time scales shorter than one second [3]. Studies have focused on four main tasks to elicit cognitive load - object manipulation, reading comprehension, mathematical reasoning and searching [4]. These areas are quite similar to the types of tasks a programmer must do, that is a programmer must set up the program in a logical structure, read previous code written and complete their program. The above four tasks are all included in programming comprehension.

1.2 Programming comprehension and cognitive load

It is well established that learning to program is difficult for many students [5]. Research largely indicates that although syntax can be problematic the most significant problem is being able to break down a problem into its component parts and express the component parts in programming code [6] [7] . Programming, like problem solving, relies heavily on working memory, where only a few items can be stored and even fewer can be processed and thus this can lead to the high levels of overload. However, very few studies have attempted to examine this. The most closely related work in this area found was a study that attempted to reduce cognitive load while learning to program by breaking down the program into a concept map to allow the programmer to visualise the code in a more graphical manner. The method was found to reduce cognitive load by administrating the NASA-TLX survey to the student [6].

1.3 Task-evoked pupillary response

When someone performs a task such as recallinginforma-

*Senior / Corresponding Author

tion from memory, paying close attention or just thinking hard their pupils tend to dilate slightly [8].After a few seconds of completing the task, the persons pupils' return to their normal state [8] [9] [10]. This response is called Task-Evoked pupillary response (TERP). This response is involuntary and associated with a broad set of cognitive functions [10] (doing mental arithmetic [11], sentence comprehension [12] and letter matching [13]). As novice programmers will do all of these cognitive functions while programming, TERP would appear to be an appropriate measure.

2. PROPOSED METHODOLOGY

This paper proposes to develop a model to describe the relationship between cognitive load and programming comprehension.
The goal of this experiment is to answer the following research question:

- How well does pupil size correlate with the mental workload that programming imposes on a student?

As the focus of this study is on novice programmer's key topics within a first year third level introductory to programming course will be used. Students undertaking their first year of study in a computing degree (that involves programming) will be invited to participate in the study. A set of tasks will be prepared based on core concepts from the first year course. The student will be asked to determine the outcome of a given code snippet. Each code snippet will contain only one core concept of programming (if statements, while loops, for loops etc). An equal number of easy and difficult programming tasks will be created. At the start of the experiment the participant will be asked to fixate on a particular spot on the screen. This will give a baseline reading for the participant. After the participant has read and understands the instructions they will be begin by completing practice trials. Then the participant will begin the real trials. After completing the trial, a NASA-TLX survey will be given in order for the students to gauge their own perceived cognitive load.

2.1 Measurements

Along with the NASA-TLX survey, the participants pupil data, on-screen activities and time completion will be recorded. The percentage change in pupil size(PCPS) will be computed for each instance. The average PCPS will then be computed over the entire experiment. Inter-and intra- participant eye gaze patterns will be examined and evaluated.

3. CONCLUSIONS

The basis of this model is heavily grounded in research conducted in cognitive workload in Human-Computer Interaction. This model however builds on that research and attempts to build a relationship between cognitive load and programming comprehensions. Novice programmers are of particular interest in this study. If the level of understanding of core concepts in first year computer programming can be increased then the level of continuing students may increase.

4. ACKNOWLEDGEMENTS

This work was funded by the Irish Research Council Postgraduate Scholarship 2015.

5. REFERENCES

[1] Dale Shaffer, Doube Wendy, and Juhani Tuovinen. Applying Cognitive Load Theory to Computer Science Education. *Joint Conf. EASE & PPIG*, (April):333–346, 2003.

[2] Eija Haapalainen, SeungJun Kim, Jodi F. Forlizzi, and Anind K. Dey. Psycho-Physiological Measures for Assessing Cognitive Load. *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 301–310, 2010.

[3] Jeff Klingner. Measuring cognitive load during visual tasks by combining pupillometry and eye tracking. *Perspective*, (May):130, 2010.

[4] Muhammed Yousoof and Mohd Sapiyan. Cognitive Load Measurement in Learning Programming Using NASA TLX rating scale. pages 235–245.

[5] Susan Bergin and Ronan Reilly. Programming: factors that influence success. *ACM SIGCSE Bulletin*, 37:411–415, 2005.

[6] Muhammed Yousoof, Mohd Sapiyan, and Khaja Kamaluddin. Reducing cognitive load in learning computer programming. *World Academy of Sciene, Engineering and Technology*, 12(March):469–472, 2005.

[7] Jackie O'Kelly, Susan Bergin, S. Dunne, Peter Gaughran, John Ghent, and Aidan Mooney. Initial findings on the impact of an alternative approach to Problem Based Learning in Conputer Science. *Pleasure by Learning*, 2004.

[8] Shamsi T Iqbal, Xianjun Sam Zheng, and Brian P Bailey. Task-evoked pupillary response to mental workload in human-computer interaction. *Extended abstracts of the 2004 conference on Human factors and computing systems CHI 04*, page 1477, 2004.

[9] Shamsi T Iqbal and Pd Adamczyk. Towards an index of opportunity: understanding changes in mental workload during task execution. *Conference on Human Factors in Computing Systems*, pages 311–320, 2005.

[10] Jeff Klingner, Rakshit Kumar, and Pat Hanrahan. Measuring the Task-Evoked Pupillary Response with a Remote Eye Tracker. *Proceedings of the 2008 symposium on Eye tracking research & applications.*, 1(212):69–72, 2008.

[11] Eckhard H. Hess. Attitude and pupil size. *Scientific american*, 1965.

[12] Marcel A. Just and Patricia A. Carpenter. The intensity dimension of thought: Pupillometric indices of sentence processing. *Canadian Journal of Experimental Psychology*, pages 310–339.

[13] Jackson Beatty and Brennis L Wagoner. Pupillometric signs of brain activation vary with level of cognitive processing. *Science (New York, N.Y.)*, 199(4334):1216–1218, 1978.