Reading skills can predict the programming performance of novices: an eye-tracking study

Marthie Schoeman
Unisa
schoema@unisa.ac.za

DOI: http://dx.doi.org/10.18820/2519593X/pie.v37i2.3

ISSN 0258-2236

e-ISSN 2519-593X

Perspectives in Education

2019 37(2): 35-52

Date Published:
27 November 2019

Published by the UFS
http://journals.ufs.ac.za/index.php/pie

ABSTRACT

Due to the character of programming languages, reading ability may have more impact on learning to program than on learning in other subjects. This paper describes an exploratory study of the relationship between reading skills, as perceived through eye tracking, and the ability to program. An empirical investigation into this relationship determined that students with inadequate reading skills are at risk of failing at introductory programming. As an explanation for the effect of reading ability on learning to program, we argue that a programming language is a special high-level written language and that using it requires high levels of comprehension, inferencing, selective attention, organising and reflecting. As a result, a student's reading ability will have a considerable effect on learning to program. Lack of reading skills may therefore be a factor that affect students' ability to learn to program. Eye tracking can expose reading skills and, therefore, be used to identify at-risk introductory programming students. The practical contribution of this research is the demonstration of how eye tracking can reveal reading problems among programming students. We relate these reading problems to their programming performance, providing a theoretical account of the connection. The results suggest that efforts to improve reading skills could have a positive impact on learning to program.

Keywords: *Human-computer interface; interactive learning environments; post-secondary education; programming and programming languages; reading skills; teaching/learning strategies*

1. INTRODUCTION

A considerable amount of computing education research concentrate on the challenges novices experience in learning to program, the reasons therefor and aspects that forecast student success. However, a programming language is similar to other languages; with the difference being that it is never spoken, only written. Furthermore, it is a high-level language that requires the capacity to infer meaning between text elements and relate information between textual units that do not always appear in sequence. This suggests that the ability to use a programming language proficiently would necessitate good reading ability.

The relationship between students' reading ability, as revealed through eye tracking, and their experience of learning to program is considered in this paper. We discuss the relationship between reading patterns, as revealed by means of eye tracking, and students' performance as novice programmers. These eye-tracking observations were part of a larger project to develop a program visualisation (PV) tool to teach introductory programming students in an ODL environment, to trace programs in order to improve students' programming skills. The research question for the larger project was "Will a tutorial to teach tracing to visualise program execution and parameter transfer assist students in acquiring programming skills in an ODL environment so that it is noticeable in average marks?". The purpose of the eye-tracking study was to evaluate the usability of the PV artefact developed to assist learner programmers. During the eye-tracking evaluation, the relationship between students' gaze patterns, as captured while reading text on screen, and their performance as novice programmers emerged. This led us to believe that the students' reading skills have a significant impact on their capacity to learn how to program, and that using eye tracking to detect reading patterns could offer a means to identify at-risk students in introductory programming modules. The main contributions of this article to this discipline are:

- an explanation of why lack of reading skills may have an effect on students' capacity to learn to program
- empirical evidence confirming a relationship between reading proficiency and learning to program
- a demonstration of how eye tracking can be used to expose poor reading skills so as to identify students at risk of failing.

In the sections to follow, we first present a literature review that covers the process of reading, how eye tracking can be used to reveal reading ability, the relationship between reading and academic success and the relationship between reading program code and learning to program. The current situation is considered briefly. We discuss two cycles of direct live observation and eye tracking, as well as an analysis of participants' gaze replays and their responses to a questionnaire. We then relate their achievement in an entry-level programming course to the results, with possible explanations for the major effect inadequate reading skills seems to have on learning to program.

Ethical clearance for this study has been granted by the Research and Ethics Committee (CREC) of the College of Science, Engineering and Technology's (CSET) of Unisa.

2. A REVIEW OF RELATED WORK

In this review, we relate reading ability to learning by describing the cognitive processes involved in reading to learn. We also describe how reading skills can be detected by eye tracking. Finally, we review the relationship between academic success and reading as well as the role of reading in learning to program.

2.1 Reading and learning

Reading is a cognitive-linguistic activity representing a combination of several skills of which decoding and comprehension are the two primary elements. Decoding refers to the more technical ability to convert written symbols into language. Comprehension concerns creating meaning from the entire text (Pretorius & Ribbens, 2005). Good decoding skills are necessary for effective comprehension, however good decoding skills on its own will not suffice for successful comprehension (Bertram, 2006). To successfully comprehend, requires inferencing too, i.e. the capability to complete gaps between text elements and relate information across textual units (Rickheit, Schnotz & Strohner, 1985). According to Kendeou, Van den Broek, Helder and Karlsson (2014:6), “at its core, reading comprehension involves the construction of a coherent mental representation of the text in the reader’s memory”. To be a good reader requires being a good comprehender (Bertram, 2006).

Reading also includes additional higher-level executive function processes like organising and contemplating information in working memory while selectively focussing on particular factors in the text (Kendeou *et al.*, 2014). According to the cognitive load theory, to learn requires processing information in both working memory and long-term memory. Working memory has a restricted processing capacity, constrained by the intrinsic cognitive load (the difficulty of the subject matter), extraneous cognitive load (the way information is presented) and germane cognitive load (when information is presented in such a way that it facilitates the construction of cognitive schemas) (Wouters, Paas & Merriënboer, 2008).

The difference between decoding and comprehension allows for the identification of three levels of reading ability, each reflecting a specific reading maturation level (McCormick, 1997):

- Independent level: Reading with 98% decoding accuracy and a 95% level of comprehension enable such highly-skilled readers to access information independently and learn from texts effectively.
- Instructional level: Readers reading with 95% decoding accuracy and a 75% level of comprehension do not have major reading problems, but reading instruction at their development level will benefit them.
- Frustration level: Readers with less than 90% decoding accuracy and 60% or less comprehension have major reading problems, particularly with comprehension, and read below maturation level. They need intensive reading programmes to improve their reading ability.

Decoding allows children to *learn to read*, but they need to understand (comprehend) to *read to learn*. During Grades 9 to 12, successful readers master demanding subjects better through reading than by listening, whereas listening comprehension is better than reading comprehension for less skilled readers. At the post-school stage, highly-skilled readers find reading more effective than listening when learning (Pretorius & Ribbens, 2005).

2.2 Reading skills, thinking skills and eye tracking

Rayner and Pollatsek (2006) consider capturing eye movements as one of the best techniques to investigate language comprehension processes. Recording eye movements also allows studying cognitive processes and cognitive load (Zagermann, Pfeil & Reiterer, 2016). Eye movement patterns are associated with reading skills (Ashby, Rayner & Clifton, 2005; Chace, Rayner & Well, 2005; Krieber, Bartl-Pokorny, Pokorny, Einspieler, Langmann, Körner, Falck-Ytter & Marschik, 2016). Programme for International Student Assessment (PISA) results distinguishing between low- and high-scoring readers also correspond with eye-tracking measurements of reading skills (Krstić, Šoškić, Ković & Holmqvist, 2018).

Eye movements during reading comprise fixations (where eye movement pause on a particular zone), saccades (moving the eyes between fixations) and regressions (saccades where eyes are moving backwards in the text). Regressions typically occur when a reader finds the text difficult or do not understand what was read (Rayner & Pollatsek, 2006). Regressive eye movements give an indication of the point at which a reader realises (if unconsciously) that the initial decoding was incorrect (Frazier & Rayner, 1987; Rayner & Pollatsek, 2006). The level of difficulty associated with understanding a word during reading determines how long readers fixate on that word. Readers will fixate longer on less frequently used and less predictable words, skip these words less and refixate on them more often (Rayner & Pollatsek, 2006). A high fixation duration with a lower fixation rate indicate increased strain on the working memory (Chen, Epps, Ruiz & Chen, 2011).

Both fixation and saccade length are influenced by text difficulty, reading skills and characteristics of the writing system and can cause substantial variation in these values. The cognitive workload required in linguistic processing determines to a large extent the duration of fixations (Radach & Kennedy, 2013). The length and velocity of saccades are also affected by cognitive workload (Zagermann *et al*., 2016). Longer fixations, shorter saccades and more regressions are characteristic of beginner, less skilled and dyslexic readers, and characterises reading more difficult text too (Rayner, 2009). Non-English-speaking readers have longer fixations and shorter saccades, resulting in a decreased reading speed when reading English text (Beymer, Russell & Orton, 2008). To dyslexic readers, letters may appear to move around so that their vision is unstable (Stein & Walsh, 1997). Mindless reading shows longer and more off-text fixations (Reichle, Reineberg & Schooler, 2010). Displays of rapid disruptions in eye movement control may signify difficulties in processing syntax (Frazier & Rayner, 1987).

In summary, eye-tracking studies show that skilled readers display shorter fixation durations, with fewer fixations and regressions (Krieber *et al*., 2016). Poor or low-skilled readers use longer fixations, different saccadic patterns and more regressions, and they also focus less on relevant text and reread text unselectively (Kaakinen, Lehtola & Paattilammi, 2015).

2.3 Reading and academic success

Marshall (2016) considers reading skills to be inextricably linked to academic success. Poor reading skills can account for up to 80% of all learning difficulties (Louden, Chan, Elkins, Greaves, House, Milton, Nichols, Rivalland, Rohl & Van Kraayenoord, 2000) and reading ability is viewed as one of the most important predictors of academic success in higher education (Bertram, 2006; Marshall, 2016).

Reading ability is also a predictor of understanding and learning science (Hall, Maltby, Filik & Paterson, 2016). Ritchie and Bates (2013) found that efficient childhood mathematics and reading skills are associated with higher marks as well as higher qualifications and higher socioeconomic status.

The effect is more pronounced for second-language English speakers having to read English text (Gous & Roberts, 2014). Our study was conducted in South Africa, where, for most university students, English is their second language. Pillay and Jugoo (2005) observed that South African programming students whose first language differed from the teaching language achieved lower marks. Stoffelsma and Spooren (2018), too, found that students' academic English reading proficiency at the start of their tertiary education in a non-Western multilingual academic context (Ghana) predicted and affected both their marks in the first year and the fourth year.

2.4 Reading and learning to program

An absence of viable mental models for programming is often the cause for novice programmers' problems (Kieras & Bovair, 1984). Kieras and Bovair define a mental model as some kind of understanding of how a device works in terms of its internal structure and processes. A model is a representation of some information, physical reality or virtual entity that hides details and aspects not critical to the phenomenon being modelled. It can therefore be manipulated and simulated, but still has the ability to provide predictions or cause change in the modelled world (Isbell, Stein, Cutler, Forbes, Fraser, Impagliazzo, Proulx, Russ, Thomas & Xu, 2010). A mental model is thus a person's internal representation of an entity, how it behaves and what responses can be expected from it.

Mental models for programming include models of the programming language (i.e. control, data structures and data representation) and the computer, an understanding of the real-world problem domain and program design as well as a model of the actual program (Robins, Rountree & Rountree, 2003). Novices have difficulty in basic program planning and they often have fragile programming-specific knowledge (Pears, Seidman, Malmi, Mannila, Adams, Bennedsen, Devlin & Patterson, 2007; Robins *et al*., 2003), but learning to *apply* rather than *understand* basic concepts appears to be the biggest problem (Dale, 2006; Lahtinen, Ala-Mutka & Järvinen, 2005; Milne & Rowe, 2002; Soloway & Spohrer, 1989). One can only apply knowledge once it has been internalised, which requires a viable mental model of the knowledge. Readers construct a mental model of the content of text (Kendeou *et al*., 2014). A skilled reader will be more experienced in creating mental models, which should assist in creating the mental models involved in programming.

The section of the brain where reading is processed was initially used to find objects and circumnavigate the surroundings (Gous & Roberts, 2014). Like navigation, reading consists of negotiating objects (words and letters) in a specific setting (the larger text). Similarly, program code represents an abstract space to be surveyed with the same cognitive approaches as used to investigate natural surroundings (Reddivari & Kotapalli, 2017), since to understand a program is a "process of generating a cognitive map from the program source-code to a set of application domain concepts" (Cox, Fisher, & O'Brien, 2005:93). This corresponds to developing a mental model. Students' spatial ability, i.e. their capability to circumnavigate, and their marks in programming modules display a high correlation, in contrast to far lower correlations between their spatial ability and their marks for non-programming modules. Students with a low spatial ability also used less effectual navigational approaches in their attempts to understand code (Jones & Burnett, 2007). The association between reading and navigational ability accounts for the higher degree of problems students with lower spatial ability experience with programming, sustaining Chmura's finding (1998) that students lacking in comprehension skills find it difficult to learn to program. Greater spatial skills also provide for more effective understanding of science texts, apparently because of the part that spatial skills play to create mental models (Jaeger, Taylor & Wiley, 2016).

The BRACElet project (Lister, Fidge & Teague, 2009; Venables, Tan & Lister, 2009) found that students who cannot read and explain program code cannot write program code. In this instance, reading code refers to tracing code, not to reading natural language text, as discussed earlier. Tracing is the "mental simulation of a program execution" (Vainio & Sajaniemi, 2007:78), that is, to imitate, at some level of detail and accuracy, the process of a computer executing code (Fitzgerald, Simon & Thomas, 2005). This involves creating a mental model of what the code does, which requires comprehending and inferencing on a higher level than reading natural language text.

To relate this to the reading maturation levels discussed in section 2.1, we can infer that in order to trace and explain program code, a student needs to be at least at instructional level, so that a mental model can be created. Readers at frustration level will struggle to achieve this.

Furthermore, the tight integration of concepts in a programming language signifies a heavy intrinsic cognitive load (Robins, 2010). Insufficient reading skills would enforce a heavy cognitive load on working memory, hindering inferencing, paying selective attention and organising and reflecting, thus increasing the difficulty in understanding the study material. Students who experience reading problems accordingly will find it difficult to learn to program. Busjahn, Schulte and Busjahn's study (2011), comparing natural language text reading and code reading, also found a significant increase in fixation duration and regression when subjects read source code, confirming the higher level of difficulty required to read and understand source code.

3. CURRENT SITUATION

A systematic literature review on introductory programming between 2003 and 2017 inclusive reveals that a small number of papers (no more than three per year) report on studies on the thinking processes of novice programmers (Luxton-Reilly, Simon, Albluwi, Becker, Giannakos, Kumar, Ott, Paterson, Scott, Sheard & Szabo, 2018). Nonetheless, the use of eye tracking in programming to study cognitive processes in programming is rising (Obaidellah, Al Haek & Cheng, 2018). These studies are typically qualitative involving small numbers of participants, justifying larger and quantitative studies on the thinking processes of novice programmers.

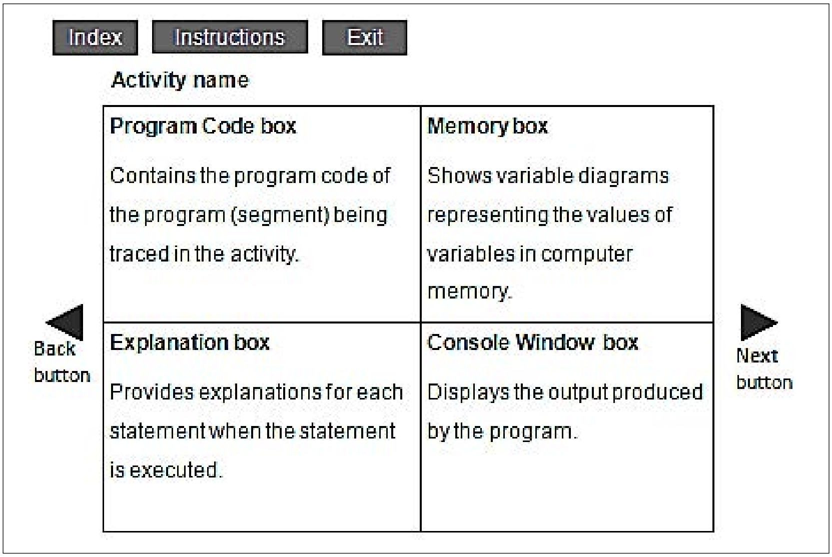
Since the ability to read program code has been identified as a fundamental building block in learning to program (Lister *et al*., 2009; Venables *et al*., 2009), some work has been done on assisting students in developing code reading skills (Sudol, Stehlik & Carver, 2012; Busjahn & Schulte, 2013; Cunningham, Blanchard, Ericson & Guzdial, 2017). However, according to Obaidellah *et al.* (2018), by June 2017 only two papers on comparing the difference between reading natural language text and source code had been published (Busjahn *et al*., 2011; Peachock, Iovino & Sharif, 2017); reading is often overlooked in the teaching of programming skills (Bednarik, Schulte, Budd, Heinemann & Vrzakova, 2018) and the process of reading code is not well understood either (Luxton-Reilly *et al*., 2018).

4. CONTEXTUALISATION OF THE STUDY AND METHODOLOGY

4.1 Background

This exploratory study stems from a larger design-based project aimed at improving the tracing skills of first-year programming students enrolled for an introductory C++ programming course. For this purpose, an interactive computer-based artefact to teach students to trace program code by drawing variable diagrams was developed and tested. The research was done over four semesters at Unisa – a large open distance and e-learning (ODeL) institution where, during the study, between 1 200 and 2 200 students were enrolled per semester. We evaluated the artefact, called the Drawing Variable Diagrams Tutorial (DVDT), for usability and user experience (UX) using eye tracking and a post-test questionnaire. Usability is defined as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (ISO 9241-210:2010(E), 2010:3). User experience is related to usability and defined as "a person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service” (ISO 9241-210:2010(E), 2010:3).

The DVDT is intended to assist students in learning how to trace program code by following the process of doing so. Each activity in the DVDT demonstrates the tracing of a program or program segment by drawing variable diagrams, statement by statement. During the program trace, each statement is also explained. To explain a statement, the screen is divided into four panels, as shown in Figure 1. The four panels respectively show the program code, the computer memory, the program output and an explanation of the statement. The purpose of each panel is briefly explained inside the corresponding box in Fig. 1. The “Next” and “Back” buttons is used to move forwards and backwards through an activity. The “Index” button provides a drop-down menu that allows users to choose an activity, while the Instructions button provides an explanation of how to use the tool.



Program Code box	Memory box
Contains the program code of the program (segment) being traced in the activity.	Shows variable diagrams representing the values of variables in computer memory.
Explanation box	**Console Window box**
Provides explanations for each statement when the statement is executed.	Displays the output produced by the program.

Figure 1. Screen layout for tutorial

While analysing the results of the eye tracking study, it became noticeable that certain students exhibited reading patterns indicative of insufficient reading skills. These students typically also did not pass the module. We investigated this further using the methods explained below.

4.2 Methods

The DVDT was developed and tested repeatedly over two cycles (two years). One of the methods used to evaluate the DVDT is user testing conducted in a human-computer interaction (HCI) laboratory. A usability study using eye tracking tested one set of students from each year, i.e. two sets in total. The intention was to observe and study how students would interact with the DVDT when using it by themselves and at home.

4.2.1 Data collection

Volunteers for observation and eye tracking as they used the DVDT was recruited from the course cohort using self-selecting sampling. An email was sent to all students registered for the course inviting them to participate. No reward was proffered for participating. Sixteen students in total volunteered for the evaluation.

Both the first and the second cycle of usability and UX evaluation employed the same procedure. A Tobii T120 eye tracker with a fixation radius set at 35 pixels and a 5-point eye tracking calibration was used to record participants’ eye movements while they performed various tasks on the DVDT. Students were observed one at time. Once a participant had signed the informed consent form, the purpose of the eye tracking session was explained to him/her and the calibration performed. The participants received a task list that comprised six activities they had to work through during the observation. This included both a pre-test comprehension test and a retrospective comprehension test as well as an open-ended post-test questionnaire[1] to reflect on the encounter.

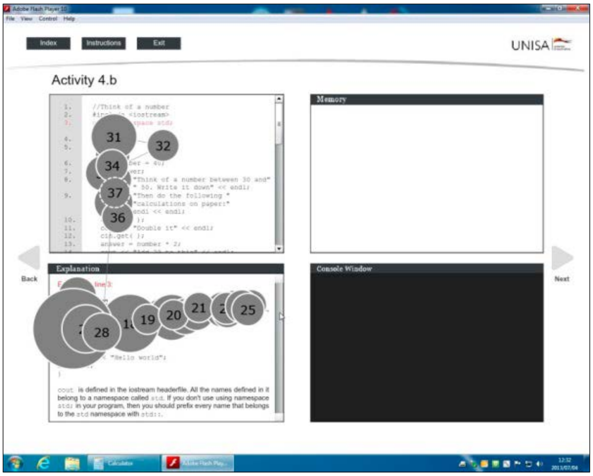
Data in the form of gaze videos (showing the animated fixation paths) and static gaze plots was exported using the Tobii Studio software. Gaze videos are graphical representations of the gaze data superimposed on screen recordings of the students’ interaction with the DVDT. The gaze videos include a small video showing the participants’ face.

4.2.2 Data analysis

The static gaze plots were not used in the analysis because they were not really comparable across participants. The reason is that participants followed diverse paths through the tasks and scrolled in different ways through the contents of the individual panels. We analysed only the gaze videos. The animated gaze videos were analyzed qualitatively by creating rich descriptions of each participant’s interaction with the tutorial. All videos were viewed individually, in succession, while detailed notes of each participant’s actions, menu selections, reactions and body language on the recorded gaze videos were made. The fixation patterns, gaze paths and regressions were also analyzed for each participant. The sequence of fixations indicated how their eyes moved between the four quadrants of the tutorial screen. The time spent reading a statement or description, and the time spent on each quadrant was also analysed.

During this analysis we were specifically looking for insight into the usability and user experience of students as they used the DVDT. A side effect of this analysis was that students’ reading ability became apparent through the results. Fig. 2 and Fig. 3 show extracts from the gaze paths of a skilled reader and a participant less skilled in reading respectively. The sequence of fixations is indicated by the numbers on the fixation circles. In Fig. 3 the fixations jump from panel to panel while in Fig.2 the fixation path follows the words as they are read.

[1] The questionnaires are available on request from the first author or can be found in Schoeman(2016).



Activity 4.b

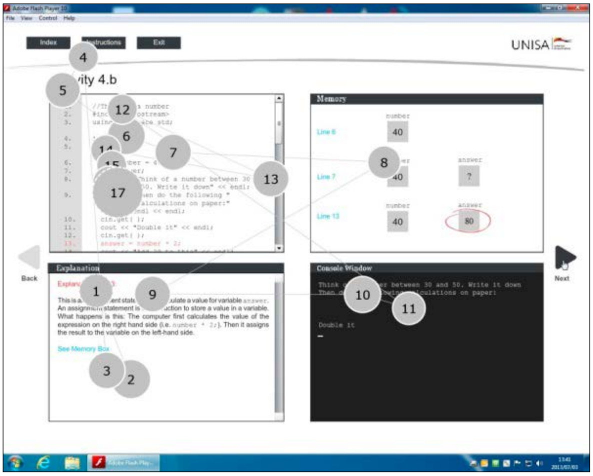
```
//Think of a number
#inc    <iostream>
         space std;

        ber = 40;
        wer;
        "Think of a number between 30 and"
        " 50. Write it down" << endl;
        "Then do the following "
        "calculations on paper:"
        endl << endl;
        );
  co    "Double it" << endl;
  cin.get( );
  answer = number * 2;
```

line 3:

```
      << "Hello world";
}
```

cout is defined in the iostream headerfile. All the names defined in it belong to a namespace called std. If you don't use using namespace std; in your program, then you should prefix every name that belongs to the std namespace with std::.

Figure 2. A typical gaze path of a participant with good reading ability



ity 4.b

```
//Th         a number
#inc         ostream>
using        ace std;

        ber = 4
        wer;
        Think of a number between 30
        50. Write it down" << endl;
        en do the following "
        alculations on paper:"
        ndl << endl;
  cin.get( );
  cout << "Double it" << endl;
  cin.get( );
  answer = number * 2;
```

	number	answer
Line 6	40	
Line 7	40	?
Line 13	40	80

Explan   3:

This is a   ent state   ulate a value for variable answer. An assignment statement is   uction to store a value in a variable. What happens is this: The computer first calculates the value of the expression on the right hand side (i.e. number * 2;). Then it assigns the result to the variable on the left-hand side.

See Memory Box

```
Think o      er between 30 and 50. Write it down
Then d          culations on paper:

Double it
_
```

Figure 3. A typical gaze path of a participant with poor reading skills

Analysing the pre-test and retrospective comprehension test results gave an indication of the level of learning attained from using the DVDT, while the students' UX and usability problems were revealed in the post-test questionnaire.

5. RESULTS OF THE EYE-TRACKING STUDY

We discuss the results between the two cycles independently. After the cycle 1 evaluation, the DVDT interface was updated and we can therefore not combine or compare results across the two cycles.

5.1 Cycle 1 results

Ten participants volunteered for the first cycle's eye tracking study: two females and eight males. The ages of the participants varied from 19 to 33. They speak eight different home languages. Four participants were registered for the course for the first time, while the others were all repeating the course, although they did not all progress to writing the examination previously. All maintained that they were computer literate.

Four participants passed the final examination, three failed and two did not write it. One female had a vision impairment and her eyes could not be tracked successfully. Her data was not included in the analysis. Participants' interaction with the DVDT and their experience of the usability of the DVDT were investigated by examining the gaze replays and observation of the videos accompanying the gaze replays. Four main concerns were distinguished:

- Students experienced problems with learning from the tutorial,
- Students found the instructions for using the tutorial inadequate,
- Students found it difficult to navigating through the tutorial and
- Some students displayed poor reading skills.

This paper focus on the reading skills of participants.

Four participants (P1, P2, P3 and P7) demonstrated eye movements that suggested good reading skills. P1 and P2 read at an acceptable rate and displayed no noticeable reading difficulties. P3 read well and faster than all the other participants. P7 read at an acceptable rate and had no problems in reading and comprehending what he had read, since he followed instructions without faltering. These participants all passed the final examination at the end of the semester.

Participants P4, P5, P6, P8 and P9 all exhibited eye movements indicative of patterns associated with lack of reading skills, such as erratic eye movements (P4, P5 and P6), rereading text recurrently (P5, P8 and P9), short saccades and long fixations (P8 and P9) or using the mouse to guide reading (P6). P4 sometimes appeared mystified. Participants P6 and P9 were absent from the examination, while the others all failed the examination. We elaborate on their results below.

On P4's facial video he occasionally seemed puzzled. His gaze pattern confirms this – he tried to follow, sometimes fixating on code, or reconciling the content of the Memory box with the program code, but did not always manage. His eye movements in the Explanation box were erratic, resembling scanning, or displaying a zigzag pattern with frequent regression. From this we deduce that he probably did not understand all the explanations. It is possible that he became tired or lost concentration, resulting in mindless reading but it could also have occurred due to syntactic processing difficulties.

P5 recommended adding "a zooming tab for those who have difficulties with reading small or medium texts". The participant had a reasonable reading speed, but sometimes his fixations jumped around and he reread sentences, seemingly trying to understand. Overall his gaze patterns indicate that he found the text difficult to comprehend.

P6 did not fixate sufficiently on the Explanation box to support comprehension. There is some indication of reading problems since he would read one or two lines of the explanations and then veer off to try something else. He also used the mouse to guide his reading. This created the impression that he did not understand or was not a skilled reader.

P8 read slowly, reread sentences and seemed to struggle to comprehend. His eye movements showed short saccades and long fixations which indicate that either he found the text difficult or was a beginner or dyslexic[2] reader (Rayner, 2009).

P9 was also a poor reader, but his actions differed from that of the others. He went through the activities very quickly, sometimes barely reading anything at all. When he did read, his eye movements often showed fairly short saccades and long fixations, and he sometimes reread text. When he got stuck, he either just clicked the "Next" button without reading or used the "Back" button and the "Index" button to exit from the activity. This could be an indication of poor reading skills, but it could also just be that he was affected by the laboratory setting and could therefore not perform the tasks properly. This participant was absent from the examination.

Two questions in the open-ended post-test questionnaire that students completed after the eye tracking and observations in the HCI laboratory, can shed light on students' reading skills:

- "Would you like to have sound / someone explaining how to do it while using the tutorial?"
- "What would you change?"

Four participants requested auditory explanations; one would prefer to have the option of auditory explanations on request, and four favoured using the tutorial without them. Of the four participants who preferred auditory explanations, two did not pass the examination (P4 and P8), and the other two were absent in the examination (P6 and P9). They all seemed to be lacking in reading skills. Three of the four participants who favoured no auditory explanations passed the examination and displayed good reading skills (P1, P3 and P7). Participant P5 did not favour auditory explanations, but experienced difficulty in comprehending and failed the examination. Participant P2 favoured occasional auditory explanations, although he could read well and passed the examination. The feedback may be an indication that those with lesser reading skills believed auditory explanations would compensate for their lack in reading skills.

The analysis of the results for the first cycle is summarised in Table 1.

Table 1. Summary of results for first cycle

Participant	Reading skills	Auditory explanations requested	No. of times registered for the course	Exam results
P1	Good	No	2	50%
P2	Good	Occasional	1	61%
P3	Good	No	1	64%
P4	Lacking	Yes	1	39%
P5	Lacking	No	5	20%
P6	Lacking	Yes	2	n/a (absent)
P7	Good	No	2	64%
P8	Lacking	Yes	2	10%
P9	Lacking	Yes	1	n/a (absent)

[2] Dyslexic readers have long fixations and short saccades with many regressions. See section 2.2.

5.2 Cycle 2 results

Following the first implementation and testing cycle, the DVDT was adapted based on the findings of the usability evaluation. A second group of participants was recruited to participate in a similar usability evaluation in the second implementation cycle. Seven participants volunteered, comprising of one female and six males. The ages of the participants varied between 19 and 31 years. The participants speak four different home languages. Five were registered for the course for the first time. One of the remaining participants was repeating it for the fifth time, while the other one did not progress to write the examination during his previous registration. All maintained that they were computer literate. Three participants passed the final examination, three failed and one did not progress to the examination.

Similar to the first implementation cycle, gaze replays and observation were used to investigate participants' experience of the usability of the DVDT, after the modification for the second cycle. We again focus on the reading skills. The conduct and eye movements of P2, P4 and P7 indicate that they can read well and they also encountered no difficulties in using the tutorial. These three students passed the final examination at the end of the semester.

P5 also seemed to read well, judging by the speed of his eye movements and the time taken to read the instructions, although his mathematics skills were weak. It seemed as if he found mathematical calculations difficult, since he spent considerable time to figure them out. His recommendation to incorporate the use of brackets for calculations in the tutorial since "if you make a mistake with BODMAS, you're basically out", confirms this. He did not favour auditory explanations. He did not write the examination.

Participants P1, P3 and P6 were poor readers. They exhibited lack of reading skills such as reading slowly and rereading text, or using the mouse to guide their reading (P3 and P6), appearing perplexed (P3 and P6), reading aloud (P3) and displaying long fixations and erratic saccadic eye movements (P6). None of them passed the examination. We elaborate on their results below.

P1 read quite slowly, although he really made an effort to understand, by repeatedly reading the explanations and trying to reconcile the program code, the variable diagrams and the Console Window. He clearly became tired after about 45 minutes and started paying less attention, reading less and sometimes clicking the "Next" button to continue without reading the program code or explanations.

Judging by the speed of his eye movements and his behaviour, P3 could not read well. He frequently used the mouse to guide his reading and muttered all the time, reading aloud softly. He also returned to text several times and seemed to struggle to understand, frowning and fidgeting. His questionnaire response indicates that he would prefer an auditory explanation "… to understand it more in depth. Reading it doesn't mean we understand what they say especially if you don't have a programming background". This indicates that although he could decode the text (one of the main components in reading), he could not comprehend what he had decoded (the second main component in reading). He also recommended that we provide "sound disks that explain every chapter", confirming that he struggled to understand what he read.

P6 read quite slowly, often rereading the same line with very long fixations on words or phrases. He also used the mouse to guide his reading, and sometimes displayed erratic saccadic movements. He spent minutes at a time to reconcile the program code, explanations and the contents of the Memory Box, repeatedly rereading and rechecking boxes. His facial expression seemed puzzled at times indicating that he did not understand. He did very poorly in the examination.

The same questionnaire as in the first cycle was used to obtain feedback from participants after their eye tracking and observation session.

Three participants requested auditory explanations, three preferred using the tutorial without auditory explanations and one favoured having the option of auditory explanations on request. Participants with good reading skills favoured using the tutorial without audio (P2, P4, P5 and P7), although P7 suggested a toggle switch for sound to clarify more complicated sections, i.e. having optional auditory explanations. In contrast, three of the four participants who failed the examination would have preferred to have auditory explanations (P1, P3 and P6). These three participants also experienced reading difficulties. Significantly, P3's response "Reading it doesn't mean we understand what they say especially if you don't have a programming background", indicates that although he could decode the text, he could not comprehend it.

Two of the suggestions to improve the tutorial may be related to reading difficulties: firstly, to provide CDs with explanations for every chapter in the study guide as well as summaries of selected chapters and, secondly, to include optional auditory explanations with a toggle switch for further clarification or assistance.

The analysis of the results for the second cycle is summarised in Table 2.

Table 2. Summary of results for second cycle

Participant	Reading skills	Auditory explanations requested	No. of times registered for the course	Exam results
P1	Lacking	Yes	1	30%
P2	Good	No	1	58%
P3	Lacking	Yes	1	19%
P4	Good	No	1	92%
P5	Good	No	5	n/a (absent)
P6	Lacking	Yes	2	14%
P7	Good	No (toggle switch for explanations)	1	89%

6. DISCUSSION: READING SKILLS AND LEARNING TO PROGRAM

Both the comprehension phase of reading as well as the higher executive functions of inferencing, paying selective attention, and organising and reflecting in working memory, are involved in the difficulties novices experience to develop viable mental models for programming. While programming languages may look like English, a far higher level of comprehension and inferencing is involved in understanding (reading and explaining) program code, because the higher executive functions need to be used to a far higher degree than when reading natural language text. In addition, program flow is not sequential, requiring a substantial degree of selective attention, organising and reflecting, executed simultaneously in the limited working memory. Lack of reading skills will add to the strain on the working memory. The eye-tracking study's results show that students who could read well passed, whereas those who read at frustration level did not pass. Most of those who did not pass also requested auditory explanations in the tutorial. We can infer that this is an indication that they lacked the higher levels of comprehension and inferencing required to read a program.

7. CONCLUSION

This paper describes a side effect from a larger project in which an artefact to teach students to trace program code in an entry-level programming module was developed and tested. As part of the qualitative evaluation in the study, students were eye tracked in an HCI laboratory to examine the usability of the tool and students' UX thereof. On analysing the eye-tracking results, we found that the majority of the students who displayed insufficient reading skills, failed the module. Although the trial was small, the analysis of the direct live observations and eye tracking bear out the relationship between students' reading competence and their capacity to learn to program.

The relationship between reading ability and programming can be explained by the character of programming languages. Reading and understanding program code requires high levels of inferencing, selective attention, organisation and reflecting, all at the same time. This causes a high intrinsic cognitive load that correspondingly challenges the processing capacity of working memory, which makes it more demanding for less-proficient readers to master programming. The same area in the brain process both reading and navigation, including navigating through program code, which is an intrinsic part of understanding and writing program code. This sustains our contention that insufficient reading skills can make it more demanding to learn to program. Furthermore, it points to the possibility that insufficient reading skills may impact more on success in programming modules than in other modules.

The fact that our participants, who were identified with poor reading skills during the eye tracking, failed the introductory programming module substantiates our argument that inadequate reading skills can be related to students' difficulty in learning to program. Programming lecturers need to be aware of this and will have to take into account that students' reading skills may be part of the problem, when they attempt to assist struggling students. Should the reading skills of such students be enhanced, it could possibly aid them in their endeavours to learn to program as well as raise their general academic performance.

REFERENCES

Ashby J, Rayner K & Clifton C 2005. Eye movements of highly skilled and average readers: differential effects of frequency and predictability. *The Quarterly Journal of Experimental Psychology*, 58A(6): 1065–1086. https://doi.org/10.1080/02724980443000476

Bednarik R, Schulte C, Budde L, Heinemann B & Vrzakova H 2018. Eye-movement modelling examples in source code comprehension: a classroom study. In: *Proceedings of the 18th Koli Calling International Conference on Computing Education Research*. New York, NY, USA: ACM, 1-8. https://doi.org/10.1145/3279720.3279722

Bertram C 2006. Exploring teachers' reading competences: a South African case study. *Open Learning: The Journal of Open, Distance and e-Learning*, 21(1): 5-18. https://doi.org/10.1080/02680510500467874

Beymer D, Russell D & Orton P 2008. An eye tracking study of how font size and type influence online reading. *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction.* 2: 15-18.

Busjahn T & Schulte C 2013. The use of code reading in teaching programming. In: *Proceedings of the 13th Koli Calling International Conference on Computing Education Research.* New York, NY, USA: ACM, 3–11. https://doi.org/10.1145/2526968.2526969

Busjahn T, Schulte C & Busjahn A 2011. Analysis of code reading to gain more insight in program comprehension. In: *Proceedings of the 11th Koli Calling International Conference on Computing Education Research*. Koli, Finland: ACM, 1-9. https://doi.org/10.1145/2094131.2094133

Chen S, Epps J, Ruiz N & Chen F 2011. Eye activity as a measure of human mental effort in HCI. In: *Proceedings of the 16th International Conference on Intelligent User Interfaces (IUI '11*). New York, NY, USA: ACM, 315–318. https://doi.org/10.1145/1943403.1943454

Chmura GA 1998. What abilities are necessary for success in computer science. *ACM SIGCSE Bulletin*, 30(4): 55-58. https://doi.org/10.1145/306286.306316

Chace KH, Rayner K & Well AD 2005. Eye movements and phonological parafoveal preview: effects of reading skill. *Canadian Journal of Experimental Psychology*, 59(3): 209–217. https://doi.org/10.1037/h0087476

Cox A, Fisher M & O'Brien P 2005. Theoretical considerations on navigating codespace with spatial cognition. In: Romero P, Good J, Acosta Chaparro E & Bryant S (eds), *Proceedings of the 17th Workshop of the Psychology of Programming Interest Group.* Sussex University: PPIG, 92-105.

Cunningham, K, Blanchard, S, Ericson, B & Guzdial, M 2017. Using tracing and sketching to solve programming problems: replicating and extending an analysis of what students draw. In: *2017 ACM International Computing Education Research Conference. (ICER '17).* Tacoma, WA, USA: ACM, 164–172. https://doi.org/10.1145/3105726.3106190

Dale NB 2006. Most difficult topics in CS1: results of an online survey of educators. *SIGCSE Bulletin,* 38(2): 49-53. https://doi.org/10.1145/1138403.1138432

Frazier L & Rayner K 1987. Resolution of syntactic category ambiguities: eye movements in parsing lexically ambiguous sentences. *Journal of Memory and Language*, 26: 505-526. https://doi.org/10.1016/0749-596X(87)90137-9

Fitzgerald S, Simon B & Thomas L 2005. Strategies that students use to trace code: an analysis based in grounded theory. In: *Proceedings of the first International Workshop on Computing Education Research (ICER'05)*. New York, USA: ACM, 69-80. https://doi.org/10.1145/1089786.1089793

Gous IGP & Roberts J 2014. Message in a bottle: reading from paper versus reading on screen in ODeL settings. *Mediterranean Journal of Social Sciences*, 5(23): 2616-2626. https://doi.org/10.5901/mjss.2014.v5n23p2616

Hall SS, Maltby J, Filik R, & Paterson KB 2016. Key skills for science learning: the importance of text cohesion and reading ability. *Educational Psychology*, 36(2): 191-215. https://doi.org/10.1080/01443410.2014.926313

Isbell CL, Stein LA, Cutler R, Forbes J, Fraser L, Impagliazzo J, Proulx V, Russ S, Thomas R & Xu Y 2010. (Re) defining computing curricula by (re)defining computing. *SIGCSE Bull*etin, 41(4): 195-207. https://doi.org/10.1145/1709424.1709462

ISO 9241-210:2010(E) 2010. Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems. Geneva: International Organization for Standardization.

Jaeger AJ, Taylor AR & Wiley J 2016. When, and for whom, analogies help: the role of spatial skills and interleaved presentation. *Journal of Educational Psychology*, 108(8): 1121-1139. https://doi.org/10.1037/edu0000121

Jones SJ & Burnett GE 2007. Spatial skills and navigation of source code. *ACM SIGCSE Bulletin*, 39(3): 231-235. https://doi.org/10.1145/1269900.1268852

Kaakinen J K, Lehtola A & Paattilammi S 2015. The influence of a reading task on children's eye movements during reading. *Journal of Cognitive Psychology*, 27(5): 640–656. https://doi.org/10.1080/20445911.2015.1005623

Kendeou P, Van den Broek P, Helder A & Karlsson J 2014. A cognitive view of reading comprehension: implications for reading difficulties. *Learning Disabilities Research & Practice*, 29(1): 10-16. https://doi.org/10.1111/ldrp.12025

Kieras DE & Bovair S 1984. The role of a mental model in learning to operate a device. *Cognitive Science*, 8(3): 255-273. https://doi.org/10.1016/S0364-0213(84)80003-8

Krieber M, Bartl-Pokorny KD, Pokorny F B, Einspieler C, Langmann A, Körner C, Falck-Ytter T & Marschik PB 2016. The relation between reading skills and eye movement patterns in adolescent readers: evidence from a regular orthography. *PLoS One*, 11(1):1-13. https://doi.org/10.1371/journal.pone.0145934

Krstić K, Šoškić A, Ković V & Holmqvist K 2018. All good readers are the same, but every low-skilled reader is different: an eye-tracking study using PISA data. *European Journal of Psychology of Education*, 33: 521–541. https://doi.org/10.1007/s10212-018-0382-0

Lahtinen E, Ala-Mutka K & Järvinen H-M 2005. A study of the difficulties of novice programmers. *SIGCSE Bulletin,* 37(3): 14-18. https://doi.org/10.1145/1151954.1067453

Lister R, Fidge C & Teague D 2009. Further evidence of a relationship between explaining, tracing and writing skills in introductory programming. *SIGCSE Bulletin,* 41(3): 161-165. https://doi.org/10.1145/1595496.1562930

Louden W, Chan LKS, Elkins J, Greaves D, House H, Milton M, Nichols S, Rivalland J, Rohl M & Van Kraayenoord CE 2000. *Mapping the territory--primary students with learning difficulties: literacy and numeracy*. Canberra, Australia: Commonwealth Department of Education, Training and Youth Affairs.

Luxton-Reilly A, Simon, Albluwi I, Becker BA, Giannakos M, Kumar AN, Ott L, Paterson J, Scott MJ, Sheard J & Szabo C 2018. Introductory programming: a systematic literature review. In: *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2018 Companion).* ACM, New York, NY, USA: 55-106. https://doi.org/10.1145/3293881.3295779

Marshall MB 2016. Accurate on-line intervention practices for efficient improvement of reading skills in Africa. *Universal Journal of Educational Research*, 4(8): 1764-1771. https://doi.org/10.13189/ujer.2016.040804

McCormick R 1997. Conceptual and procedural knowledge. *International Journal of Technology and Design Education*, 7(1-2): 141-159. https://doi.org/10.1023/A:1008819912213

Milne I & Rowe G 2002. Difficulties in learning and teaching programming - views of students and tutors. *Education and Information Technologies*, 7(1): 55-66. https://doi.org/10.1023/A:1015362608943

Obaidellah U, Al Haek M & Cheng PC-H 2018. A survey on the usage of eye-tracking in computer programming. *ACM Computing Surveys*, 51(1):1-58. https://doi.org/10.1145/3145904

Peachock P, Iovino N & Sharif B 2017. Investigating eye movements in natural language and C++ source code - a replication experiment. In: Schmorrow D & Fidopiastis C (eds), *Augmented Cognition. Neurocognition and Machine Learning. Lecture Notes in Computer Science*, 10284: 206-218. https://doi.org/10.1007/978-3-319-58628-1_17

Pears A, Seidman S, Malmi L, Mannila L, Adams E, Bennedsen J, Devlin M & Patterson, J 2007. A survey of literature on the teaching of introductory programming. *ACM SIGCSE Bulletin*, 39(4): 204-223. https://doi.org/10.1145/1345375.1345441

Pillay N & Jugoo VR 2005. An investigation into student characteristics affecting novice programming performance. *ACM SIGCSE Bulletin*, 37(4): 107-110. https://doi.org/10.1145/1113847.1113888

Pretorius EJ & Ribbens R 2005. Reading in a disadvantaged high school: issues of accomplishment, assessment and accountability. *South African Journal of Education*, 25(3): 139-147.

Radach R & Kennedy A 2013. Eye movements in reading: some theoretical context. *Quarterly Journal of Experimental Psychology*, 66(3): 429–452. https://doi.org/10.1080/17470218.2012.750676

Rayner, K 2009. Eye movements and attention in reading, scene perception, and visual search. *Quarterly Journal of Experimental Psychology*, 62(8): 1457-1506. https://doi.org/10.1080/17470210902816461

Rayner, K & Pollatsek A 2006. Eye-movement control in reading. In: Traxler MJ & Gernsbacher MA (eds), *Handbook of Psycholinguistics*. San Diego: Elsevier, 613-657. https://doi.org/10.1016/B978-012369374-7/50017-1

Reddivari S & Kotapalli M 2017. On the use of visual clustering to identify landmarks in code navigation. In: *Proceedings - 2017 15th IEEE/ACIS International Conference on Software Engineering Research, Management and Applications (SERA 2017)*: 219–228. https://doi.org/10.1109/SERA.2017.7965731

Reichle ED, Reineberg AE & Schooler JW 2010. Eye movements during mindless reading. *Psychological Science*, 21(9): 1300-1310. https://doi.org/10.1177/0956797610378686

Rickheit G, Schnotz W & Strohner H 1985. The concept of inference in discourse comprehension. *Advances in Psychology*, 29: 3-49. https://doi.org/10.1016/S0166-4115(08)62731-4

Ritchie SJ & Bates TC 2013. Enduring links from childhood mathematics and reading achievement to adult socioeconomic status. *Psychological Science*, 1-8. https://doi.org/10.1177/0956797612466268

Robins A 2010. Learning edge momentum: a new account of outcomes in CS1. *Computer Science Education*, 20(1): 37-71. https://doi.org/10.1080/08993401003612167

Robins A, Rountree J & Rountree N 2003. Learning and teaching programming: a review and discussion. *Computer Science Education*, 13(2): 137-172. https://doi.org/10.1076/csed.13.2.137.14200

Soloway E & Spohrer JC 1989. *Studying the Novice Programmer*. New Jersey: Lawrence Erlbaum Hillsdale.

Stoffelsma L & Spooren W 2018. The relationship between English reading proficiency and academic achievement of first-year Science and Mathematics students in a multilingual context. *International Journal of Science and Mathematics Education,* 16: 1-18.

Stein J & Walsh V 1997. To see but not to read; the magnocellular theory of dyslexia. *Trends in Neurosciences*, 20(4): 147-152. https://doi.org/10.1016/S0166-2236(96)01005-3

Sudol, LA, Stehlik, M & Carver S 2012. Code comprehension problems as learning events. In: *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '12).* New York, NY, USA: ACM, 81-86.

Vainio V & Sajaniemi J 2007. Factors in novice programmers' poor tracing skills. *SIGCSE Bulletin*, 39(3): 236-240. https://doi.org/10.1145/1269900.1268853

Venables A, Tan G & Lister R 2009. A closer look at tracing, explaining and code writing skills in the novice programmer. In: *Proceedings of the Fifth International Workshop on Computing Education Research Workshop*. Berkeley, CA, USA: ACM, 117-128. https://doi.org/10.1145/1584322.1584336

Wouters P, Paas F & Merriënboer JJG 2008. How to optimize learning from animated models: a review of guidelines based on cognitive load. *Review of Educational Research,* 78(3): 645-675. https://doi.org/10.3102/0034654308320320

Zagermann J, Pfeil U & Reiterer H 2016. Measuring cognitive load using eye tracking technology in visual computing. In: Sedlmair M, Isenberg P, Isenberg T, Mahyar N & Lam H (eds), *Proceedings of the Sixth Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization (BELIV '16).* New York, NY, USA: ACM, 78-85. https://doi.org/10.1145/2993901.2993908