# e-mentor

DWUMIESIĘCZNIK SZKOŁY GŁÓWNEJ HANDLOWEJ W WARSZAWIE
WSPÓŁWYDAWCA: FUNDACJA PROMOCJI I AKREDYTACJ KIERUNKÓW EKONOMICZNYCH


2018, nr 2 (74)

Magdalena Andrzejewska, Anna Stolińska, *The eye-tracking technique in the analysis of mechanisms for solving algorithmic problems*, "e-mentor" 2018, s. 10–18, http://dx.doi.org/10.15219/em74.1347.

# The eye-tracking technique in the analysis of mechanisms for solving algorithmic problems

*Magdalena Andrzejewska**

*Anna Stolińska**

*One of the basic objectives of education is the intellectual development of children and young people, allowing them to prepare, among others, to be able to solve problems in various areas of life. This aspiration has been expressed during the recent works on the reform of education in Poland and in the core curriculum, which was approved in February 2017. It was highlighted there that IT education offers a wide range of possibilities to achieve this goal, and the appropriate means for its accomplishment is teaching programming, understood not only as learning how to write a code, but as a process of acquiring the IT approach skills to solve problems. Such understanding of IT education is not new, for many years researches have been indicating the importance of educating this special way of reasoning, which is called computational thinking. It is well-known, however, that learning algorithms and programming bring some difficulties for students at different ages. Therefore, a number of studies have been undertaken, whose aim is not only to identify the causes of these difficulties, but also to optimize the learning process by selecting appropriate teaching strategies. With that aim in mind, researchers seek for new techniques that would expand the knowledge about how brain works and allow the insight into the neurobiological aspects of the learning processes.*

## Introduction

Computational thinking is not as much associated with the ability to use ready-made algorithms, as with the use of certain procedures during the discovery of solutions to specific problems. The new core curriculum (effective since 2017) assumes that within the framework of computer science subject students will acquire the ability to analyze and solve problems in a strict, orderly manner, while maintaining the algorithmic rules of conduct – from the specification of a problem (specifying data and results and more generally – to solving the problem), by finding and developing a solution, programming the solution, testing its correctness and making possible adjustments by using a properly selected application or programming language (Reg. MEN, Journal. from 2017, item 356, vol. 1). Importantly, this algorithmic way of thinking, which Futschek (2006, pp. 159–168) wrote about, seems to be an intermediate step to achieve computational thinking skills, in which the fundamental concepts for computer science are used to solve problems, design systems, and even help with the understanding of human behavior (Wing, 2006, pp. 35–36).

Teaching algorithmics may be one of the first steps necessary to educate an IT approach in students, in order to solve problems they encounter in different life situations. However, many studies show that not only the creation of new algorithms, but also the analysis of ready-made solutions and their implementation in a programming language are difficult tasks for students (Moström, 2011, p. 7; Govender, 2006, p. 28).

The variety of problems connected with teaching algorithmics and programming (resulting both from their universality and scope) prompts to use new research techniques, appropriate for cognitive sciences, so that the analysis of cognitive processes (e.g. observational processes, including visual attention or cognitive load) would allow to find causes that prevent or hinder the acquisition of programming skills. One of such methods, which in the recent years has been more and more often applied in research projects analyzing the process of expanding knowledge, is eye tracking – a technique involving the registration of visual activity (Lai et al., 2013).

Research conducted using eye tracking, which may apply to teaching programming, focus primarily on the mechanisms of program code analysis. Researchers most often show interest in the exploration of processes that occur during programmers' activities such as code editing (reading and writing), its compilation and debugging. The studies conducted by Crosby and Stelovsky (1990), in which they compared the eye tracking data of novices and advanced programmers, and whose aim was to check how the experience affects the patterns of processing the

* Pedagogical University of Cracow

code (*code scanning patterns*) during its interpretation, can be considered pioneering. Behavior patterns during the code analysis were also investigated by Uwano et al. (2006). They noticed that when searching for errors in most of the surveyed programs, the majority of respondents first pre-read the entire code (probably in order to understand the structure of the program), and only then their focus was placed on the selected parts of the program. Moreover, the time spent on this initial code scanning had an impact on the effectiveness of error detection. The studies conducted by Sharif et al. (2012) have led to similar conclusions and additionally they have shown that the time spent affects the visual effort (measured by the number and time of fixation) needed to indicate errors. Furthermore, they observed that the experienced programmers devoted less time, as compared to the novices, to pre-scan the program before searching for actual errors. Continuing the research on these issues, Busjan et al. (2015) showed that the analysis of a program content is a non-linear procedure, and experts read the code in a less linear way than novices. In turn, Bednarik and Tukianinen (2006) recorded the oculomotor activity of students during the analysis of Java language codes in the programming environment, which provided, apart from the code editing tools, the visualization of its execution. The aim of this experiment was to examine the dynamics of student interaction with various forms of program representation during its analysis (understanding of the operation) and to compare whether these processes are similar in people with different levels of experience. The results of eye tracking studies also show that the process of reading the source code of a program is fundamentally different from reading a natural language text, however, these discrepancies are less apparent if the program code becomes similar to the text written in natural language (Busjan et al., 2011; Binkley et al., 2013). There were also studies conducted to answer the question: how code formatting, including syntax coloring affects the efficiency of its analysis (e.g. searching for syntactic errors) and its understanding (e.g. determining the result of the program). It was found that the appropriate spatial structure of the source code allows for its faster understanding, regardless of the style of naming the identifiers (Binkley et al., 2013). It was observed that syntax coloring shortens the task execution time (Dimitri, 2015) and it was found that this effect decreases with the increase in the level of programming experience (Sarkar, 2015).

So far there are only few publications on the research of mechanisms for solving algorithmic problems with the use of eye tracking as a research technique (Andrzejewska et al., 2016). The enrichment of knowledge in this area seems to be very necessary, because the lack of control over competences related to the representation and construction of algorithms is widely recognized as the primary cause of difficulties associated with programming learning (Gomes and Mendes, 2007). Conducting research in this area is also a response to the demands contained in the already cited study by Lai et al. (2013), whose authors emphasize that educational issues such as reasoning and skills acquisition, still require exploration.

The article presents the results of authors' own research, in which the state of declarative and procedural knowledge of students, associated with the forms of presentation and analysis of algorithms, was diagnosed. In the study procedure, the eye tracking technique was used as a complementary to the traditional methods, by means of which the students' oculomotor activity was recorded while they were solving some algorithmic tasks. Detailed results of this part of the experiment have been described in some other authors' publications listed in references. In this article, the most important results of eye tracking studies were synthesized in order to comprehensively describe the visual "behaviors" of the subjects, and thus show that the eye tracking technique can be useful in research on mechanisms for solving algorithmic problems.

## Research methodology

### Aims and scope of the study

The aim of the undertaken study was to diagnose the state of declarative and procedural knowledge of students associated with the forms of presentation and analysis of algorithms. The study involved the following:

a) diagnosis of knowledge (declarative) of students on the graphic representation of algorithms in the form of so-called flowcharts based on the questionnaire,
b) two-stage study of students' procedural knowledge, consisting in checking their ability to solve algorithmic tasks. In the first stage, students' eye movements were recorded while they were working on a computer monitor, the second consisted of students solving paper-based tasks.

Declarative knowledge is understood as self-esteem formulated by students, referring to the state of understanding the construction of algorithms and components of a flowchart. Procedural knowledge is revealed by behaviors – in this case it involves the ability to solve algorithmic tasks.

The use of eye tracking technique in one of the study stages was aimed at obtaining measurement data which relate to the visual activity of the subjects, based on which an in-depth analysis of students' behavior during problem solving was performed (diagnosis of procedural knowledge).

### Procedure

Each study was conducted individually and the procedure involved three stages. In the first stage, students had discussions with the members of the research team, during which they were informed about the course of the experiment. The aim of these conversations was also to reduce the level of stress, trigger positive attitudes among the respondents and motivate them to engage in the process of solving tasks. After the initial conversation, the students filled out a questionnaire in which they assessed the state of their knowledge about the algorithms presented in a form of flowcharts and indicated the source of this information.

The second stage of the study was recorded by an eye tracker and included solving tasks presented on a computer monitor. This part of the experiment began with activities that were to ensure the correct performance of the measurements (including setting position of a responder), providing the introductory instructions and the calibration of the device. Throughout the study, all the responders were provided with the same environmental conditions, such as lighting, temperature and sound insulation. During the second stage of the study, the tasks were presented in the form of slides (in the field of sciences and natural sciences: mathematics, physics, biology and computer science), which the students were to solve without the use of writing instruments. Each task required the analysis of the content of the instruction and the corresponding illustration. The time allocated for solving tasks was not limited, each pupil worked at their own pace and the order of tasks was the same. The respondents indicated the answers with the click of a mouse and verbally provided the ones, which according to them were correct. All the tasks required computational thinking from the students, however, for the purposes of this article only those that are classic algorithmic tasks are considered.

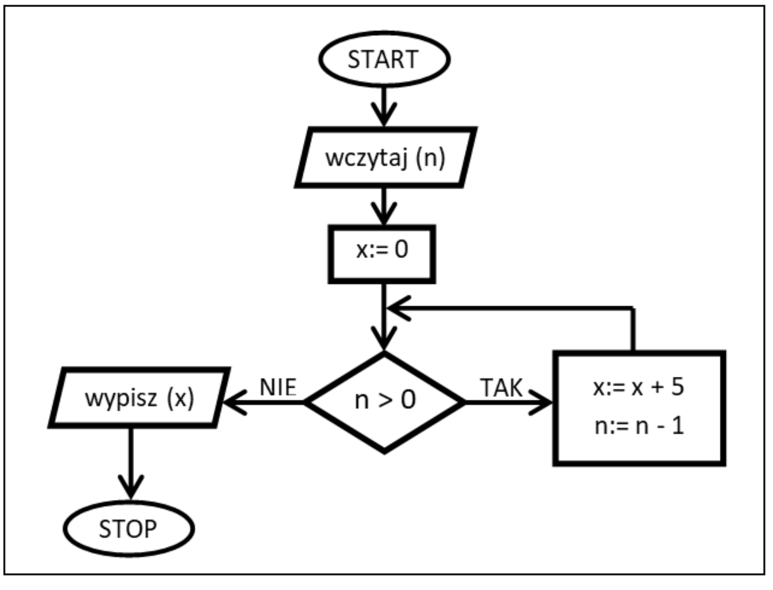
During the third stage of the study, which took place after the registration of eye tracking parameters, the students completed the questionnaire again and solved algorithmic tasks in a traditional form (in writing). Such order of research was connected with the need to minimize the impact of factors that could increase the cognitive load affecting the eye tracking indicators.

**Material**

The ALG_A task presented in figure 1 was solved by the students during the third stage of the experiment in a traditional way, using pen and paper. When determining the result, they could choose one of three ways to present it – a list of steps, a pseudocode or a flowchart. The task consisted in calculating the final value of the variable based on the analysis of the course of a loop in which this variable was modified.

The ALG_O1 (see figure 2) and ALG_O2 (see figure 3) tasks were displayed to the students on the monitor and during that time the visual activity of the study participants was recorded (the second stage). From the content of the task instruction, ALG_O1, students could read that through the analysis of the algorithm presented in the flowchart, they should indicate which set of input data will cause the algorithm not to end on the *Stop* block. The ALG_O2 task required the correct representation of the linear equation solution ax + b = 0 by the means of a flowchart. The instruction contained an analysis of the problem, so the task was to verify the correctness of the conditions stored in the appropriate blocks (conditional boxes). The ALG_O1 and ALG_O2 tasks were solved by students in black and white or color version (with so-called graphic clues).

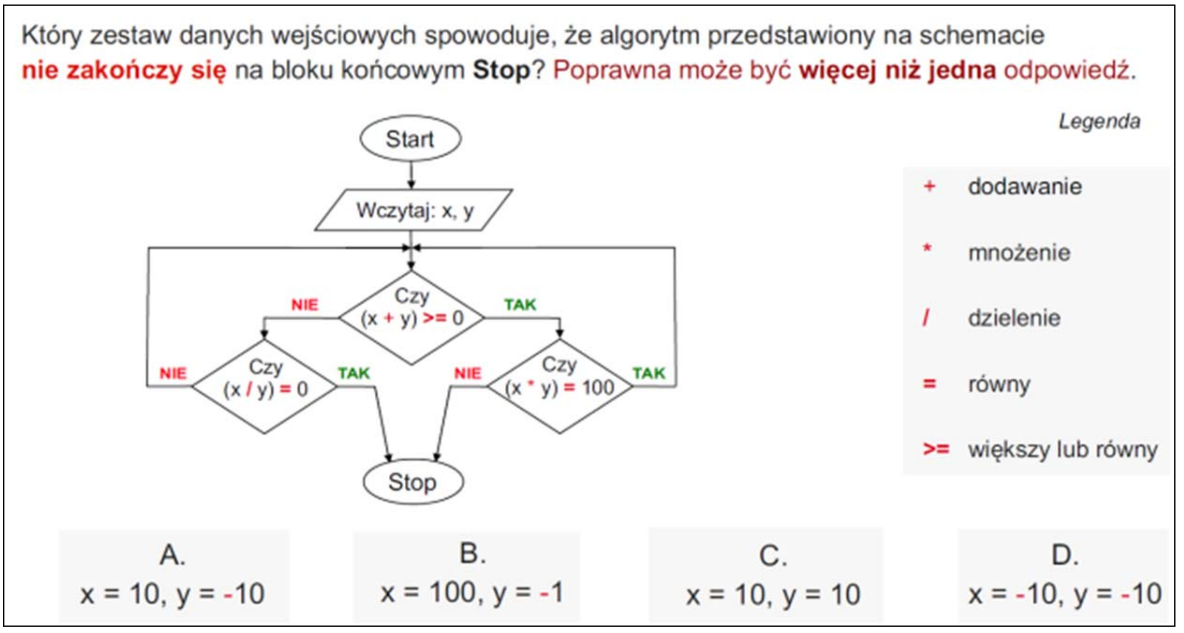
**Figure 1. The ALG_A task – flowchart version***




* In figures 1, 2 and 3 the tasks are described in Polish because this is exactly how they were presented to the students.

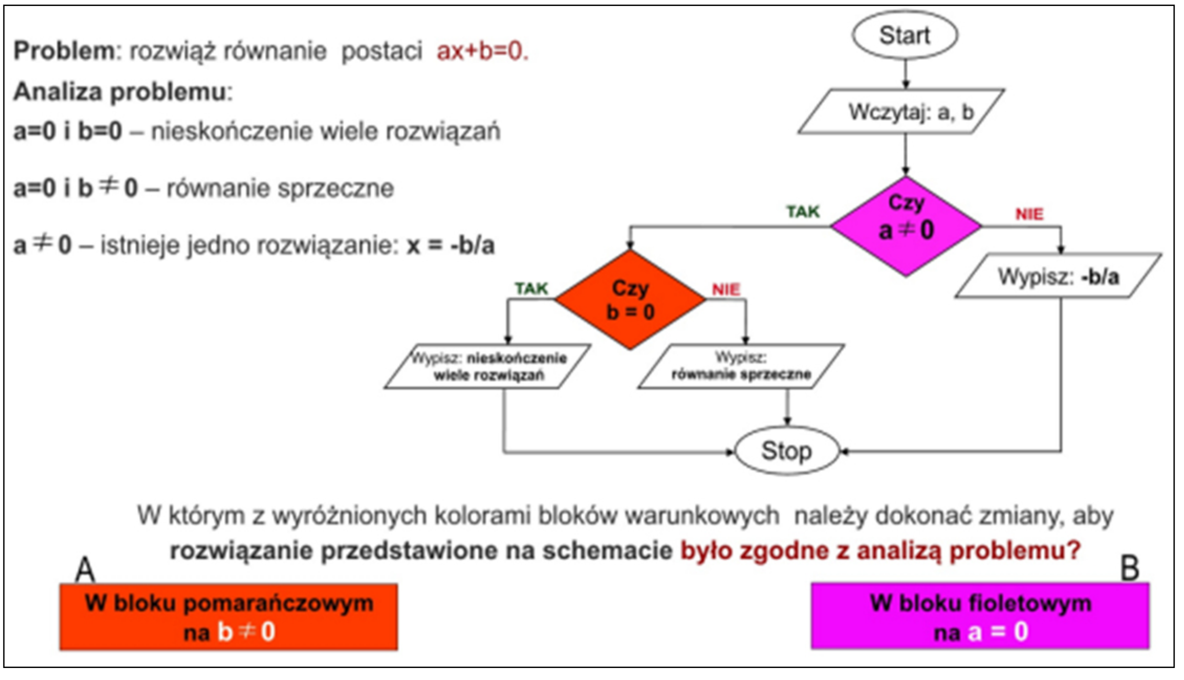
Source: authors' findings.

**Figure 2. The ALG_01 task – version with graphic clues**




Source: authors' findings.

**Figure 3. The ALG_02 task – version with graphic clues**




Source: authors' findings.

### Apparatus

In the study the SensoMotoric Instruments iViewX ™Hi-Speed500/1250 eye tracker to record data at 500 Hz was used. This eye tracking system allows to keep the head still, without limiting the respondent's field of vision. The tasks were presented on an LCD monitor with a screen size of 23" and full HD resolution of 1920x1080 pixels. The experiment was conducted using the SMI Experiment Suite™ 360 software package – the research scheme was designed in the SMI Experiment Center™ 3.4 application. The data registration was performed with the use of the SMI iView X™ module, and the results were developed using the SMI BeGaze™ 2.4 program.

**Participants**

52 third-grade secondary school students (aged 16) participated in all stages of the study, including 27 boys and 25 girls. For technical reasons, the eye data of 4 students was rejected and eventually 48 students were qualified for further analysis. The characteristics of the research group is complemented by the students' answers to the questions included in the questionnaire. The opinions analyzed below were expressed on the basis of an 11-point scale (from 0 to 10, where 0 meant very low and 10 very high). The students assessed their interest in IT as a school subject at an average level (the calculated descriptive statistics are: average M=5.83, standard deviation SD=2.93), with over 10% of respondents considering this level as low, over 50% as medium and almost 38% as high. The participants of the study were also asked whether they plan to choose a profession which will require the knowledge in the field of computer science. About 27% of respondents rated the probability of choosing such a profession as low, over 35% as medium, and almost 38% as high (M=5.42, SD=3.44). In the opinion of 4% of the respondents, IT is useful for the society at a low level, over 29% considers its usefulness as average, and 67% assesses it as high (M=7.73, SD=2.38). The results show that the students who took part in the experiment not only recognize the importance that information technology has for the development of societies, but also relatively many of them consider the possibility of acquiring professional qualifications related to this field.

## Results

**Declarative understanding of the presentation forms of algorithmic tasks and the effectiveness of their solutions**

In the first stage of the questionnaire the students read the information about the graphical presentation of the algorithm, the so-called flowcharts. This information contained a description of the purpose (function) of individual types of blocks and an example of their use. After reading the discussed description the students declared whether they are familiar with such way of representing the algorithms. For over 36% of students the principles of constructing flowcharts were completely understandable and over 63% of students declared that they understood them only partially and needed further explanations, including an example of such algorithm.

Indicating the source of their knowledge in this area, nearly 74% of respondents answered that they had become familiar with flowcharts in computer science lessons. However, a very large percentage of students (almost a quarter of them – 24%) admitted that they have not heard of this way of presenting a solution to the task, which is a worrying sign of the lack of such content implementation by teachers, as flowcharts are and have been part of the fundamental curriculum for computer science subject in secondary high schools. Detailed data is included in table 1.

The second and third stages of the study allowed to determine the procedural knowledge of students. The algorithmic ALG_A task in the questionnaire (the third stage of the study) has been presented in three corresponding forms: a list of steps, a flowchart and a pseudocode. At first, the students declared which of the recording methods is the most understandable for them (table 2), and then made a selection of one of these forms to determine the result of the algorithm for a given input value. The largest number of respondents (60%) considered the presentation of the algorithm in the form of the verbal list of steps as the most comprehensible, the flowchart or the pseudocode were indicated by about 15% of respondents, and for 10% of them all the methods were equally understandable. However, the distribution of responses related to the choice of a specific form of the algorithm presentation in order to solve the problem was different – almost 43% of the respondents made calculations based on the list of steps, 33% chose the flowchart and 24% chose the pseudocode.

In total, the percentage of correct answers was 27% (table 3), the other students either answered incorrectly (25%) or did not provide the answer (48%). The most correct answers (nearly 38%) were given by students who solved a task presented in the form of a list of steps. The level of difficulty of the task can be considered as corresponding to the standard adopted

**Table 1. Source of knowledge about flowcharts**

| Information about flowcharts is for me: | Number | % of answers | % of cases |
|---|---|---|---|
| known from IT lessons | 34 | 53.97 | 73.91 |
| known from math's lessons | 6 | 9.52 | 13.04 |
| known, because I'm interested in programming and have read about it on the Internet | 9 | 14.29 | 19.57 |
| known, because I'm interested in programming and have read about it in magazines and books | 3 | 4.76 | 6.52 |
| unknown – I have not heard about this method of presenting the solution to the task | 11 | 17.46 | 23.91 |
| Total | 63 | 100.00 | 136.96 |

Source: authors' findings.

**Table 2. The students' preferred form of the algorithm presentation**

| The most understandable form of presentation of the algorithm: | Number | % of answers |
|---|---|---|
| a list of steps | 29 | 60.42 |
| a flowchart | 7 | 14.58 |
| a pseudocode | 7 | 14.58 |
| all methods of presentation are equally understandable | 5 | 10.42 |
| none of the methods is understandable to me | 0 | 0.00 |

Source: authors' findings.

**Table 3. Effectiveness of the ALG_A task solution**

| Task solution: | Total | | List of steps | | Flowchart | | Pseudocode | |
|---|---|---|---|---|---|---|---|---|
| | Number | % ans. | Number | % ans. | Number | % ans. | Number | % ans. |
| correct | 13 | 27.08 | 7 | 38.89 | 2 | 14.29 | 3 | 30.00 |
| incorrect | 12 | 25.00 | 6 | 33.33 | 5 | 35.71 | 1 | 10.00 |
| no answer | 23 | 47.92 | 5 | 27.78 | 7 | 50.00 | 6 | 60.00 |

Source: authors' findings.

by the Central Examination Board (the comparison was made with the algorithmic task included in the extramural examination for persons who graduate in computer science at junior secondary school level). The obtained results (low effectiveness of solving the task) provide the basis for the formulation of an observation that the change in presenting students with IT knowledge is essential – the more so that according to the new core curriculum of computer science and its learning outcomes, such a task should be classified as very easy, even elementary.

As in the case of the ALG_A task, the solution of algorithmic tasks during testing with the use of the eye tracker (the second stage of research) was difficult for the students. The indicator of the correct answers for these tasks was calculated after taking into account the factor, which was the confirmation of the fact that the answer provided by them was accidental (they did not understand the task) – in the case of the ALG_O1 task 27% admitted that, whereas with the ALG_O2 task the number was 44% of students. The percentage of correct answers was 25% for the ALG_O1 task, and almost 38% for the ALG_O2 task.

In the questionnaire, the students also assessed the level of difficulty of the tasks being solved (based on the 11-point scale – from 0 to 10, where 0 – very easy, 10 – very difficult). Analyzed algorithmic tasks were classified by students as moderately difficult – ALG_O1 M=5.38, SD=2.69 and difficult – ALG_O2 M=6.35, SD=2.89. A detailed breakdown of the assessment of the difficulty level of tasks performed by students is presented in table 4. It is worth noting that among all the tasks solved by students during this study, both algorithmic tasks obtained the highest average values, i.e. they were considered the most difficult (Andrzejewska and Stolińska, 2016, p. 6).

### Synthesis of eye tracking results

The data obtained as a result of registration of students' visual activity during solving the ALG_O1 and ALG_O2 tasks (the second stage of the study) allowed for a further, in-depth analysis of the respondents' behavior during this process.

In the last few years, eye tracking studies have been increasingly recognized as a valuable research technique. On the one hand, they provide quantita-

**Table 4. Assessment of the level of difficulty of tasks ALG_O1 and ALG_O2**

| The level of task difficulty | ALG_O1 | | ALG_O2 | |
|---|---|---|---|---|
| | Number | % | Number | % |
| very easy | 4 | 8.33 | 2 | 4.17 |
| easy | 12 | 25.00 | 8 | 16.67 |
| moderately difficult | 10 | 20.83 | 10 | 20.83 |
| difficult | 17 | 35.42 | 15 | 31.25 |
| very difficult | 5 | 10.42 | 13 | 27.08 |

Source: authors' findings.

tive data related to the measurement of oculomotor parameters (such as, for example, saccades and fixations), for which statistical inference can be made; on the other hand, they allow for the qualitative assessment of results based on respondents' pathways of sight (*scan paths*) and animated films of the points in which the respondent's eyesight (*bee swarm*) is located. In addition, the separation of the so-called *areas of interest (AOI)*, together with their key indicators, allows a detailed analysis of the visual activity of the respondents. These indicators are, for example: the time that the respondents spent on observing a given area (*dwell time*) expressed as a percentage of the duration of the whole experiment, the number of fixations (*fixation count*), their average duration (*the average fixation duration*), or the number of (re-)fixations (*revisits*), as well as the time that elapses until the first fixation in a given area (*time to first fixation*) and the associated parameter indicating in which order the particular areas of interest were observed (*sequence*).

The analysis of film materials, which was supplemented by statistical calculations of numerical data, related to the visual activity in the AOIs marked on the boards enabled, in the case of the ALG_O2 task, to recognize and interpret the differences in the way the problem was solved by two selected students who obtained the correct answers (Stolińska and Andrzejewska, 2017b). The study showed different allocations of attention resources, whereas their tracking enabled the identification of a strategy that can be described as optimal, characterized by a focus of attention on the key content of the algorithm, and the control detailed analysis of the problem and multiple verification of compliance of the adopted solutions with the content of the task.

Similarly, in the case of the ALG_O1 task, based on the analysis of the scan paths, it was possible to identify the behavior of students who provided the correct answer and their classification into three groups (Andrzejewska and Stolińska, 2017). The students were divided to those who solved the task in accordance with the control resulting from conditional blocks and those who correctly indicated the answer, but mistakenly analyzed the algorithm. The third group were the respondents with "atypical" visual activity. It was also noticed that students much more frequently than needed, in the situation of solving tasks with the possibility of making calculations on a piece of paper, controlled the input data to the algorithm. In addition, the eye tracking data allowed for the detection of students who, despite providing the correct answer, made mistakes at the stage of performing mathematical operations and those who did not read the text instructions to the task.

For both tasks, the oculomotor data were subject to quantitative analysis in order to examine the differences in the basic values of the eye parameters for the two versions of the task (with or without additional graphic instructions). The aim of this analysis was to verify the views on the didactic role of such guidance also in the context of cognitive load theory (Stolińska and Andrzejewska, 2017a), which takes place in the process of learning programming. The issues related to cognitive load occurring during solving of algorithmic tasks – both internal (depending on individual properties – people's intellectual potential, their experience and other factors), as well as external ones are one of the interesting trends in the area of discussed scope of subjects. Difficulty in solving algorithmic tasks may increase the form of the presented algorithm, redundancy of information, the necessity of structuring information or making decisions with regard to the selection of content provided as a hint in the task (Garner, 2002; Stolińska and Andrzejewska, 2017a).

In the case of the ALG_O2 task, there were no significant differences in the values of the eye movement activity associated with the fixations for the two versions of this task. The result could have been influenced by the form of the hints used, which in this task were discreet in nature and neither encumber the cognitive resources of the respondents nor disturb the process of processing visual information. This result confirms the view of some researchers (Ozcelik et al., 2009) that the changes in the field of view, such as coloring or bolding the text, belong to a group of hints that cause an automatic orientation reflex in their direction. Thus, one can interpret the lack of differences in the average fixation duration and fixation count in the flowchart area in the case of the ALG_O1 task (Stolińska and Andrzejewska, 2017a) – coloring of fragments of texts in this case was not a sufficiently significant factor affecting the cognitive processes of the respondents. The positive effect of directing students' visual attention on key elements of the board, as in the case of the ALG_O2 task accompanied by the hints, also confirmed a slight shortening of its execution time and a higher percentage of the correct answers.

The analysis of the eye tracking data in the ALG_O1 task did not show that the graphic distractors attached to the task were elements significantly increasing the external cognitive load, which leads to the assumption that the students' subjective opinion about the level of difficulty of algorithmic tasks (which were considered to be the most difficult of all the tasks solved by the students) is strongly related with an internal cognitive load, which depends directly on the effort that one allocates to understand a problem (its structure and complexity).

## Summary

The low efficiency of solving relatively simple algorithmic tasks and at the same time not high level of knowledge declared by the students confirm the assumption, adopted in the new core curriculum, that the center of gravity in IT education should be focused on developing the ability to analyze and solve problems based on logical and abstract thinking. Therefore, it is necessary to further improve the knowledge about the difficulties associated with teaching algorithms and finding ways to minimize them. The use of eye tracking technique for this purpose is a promising direction for researchers' activities, and the synthesis of eye tracking results presented in the article, which provided objective information on students' behavior during solving algorithmic tasks, seems to confirm it.

The possibility of interpreting visual information processing patterns opens some new theoretical and cognitive perspectives in the field of didactics, including the methods of teaching algorithmics and programming – the areas of computer science which in essence aim to solve problems, that is, require information processing at a "higher level" and engagement of different cognitive mechanisms (attention, memory, perception). New research methods, such as eye tracking, can objectively help to explain the processes which take place during learning and to verify the teaching theory.

## References

Andrzejewska, M., Stolińska, A. (2017). Kierowanie uwagą wzrokową w procesie rozwiązywania problemów algorytmicznych. *Edukacja-Technika-Informatyka*, *2*(20), 308–314. DOI: 10.15584/eti.2017.2.40.

Andrzejewska, M., Stolińska, A. (2016). Comparing the Difficulty of Tasks Using Eye Tracking Combined with Subjective and Behavioural Criteria. *Journal of Eye Movement Research, 9*(3), 1–16. DOI: 10.16910/jemr.9.3.3.

Andrzejewska, M., Stolińska, A., Błasiak, W., Peczkowski, P., Rosiek, R., Rożek, B., Sajka, M., Wcisło, D. (2016). Eye-tracking verification of the strategy used to analyse algorithms expressed in a flowchart and pseudocode. *Interactive Learning Environments*, *24*(8), 1981–1995. DOI: 10.1080/10494820.2015.1073746.

Andrzejewska, M., Stolińska, A. (2015). Zastosowanie okulografii do identyfikacji metod analizy problemu algorytmicznego. *Edukacja-Technika-Informatyka*, *6*(3), 209–215.

Bednarik, R., Tukiainen, M. (2006). An eye-tracking methodology for characterizing program comprehension processes. *Proceedings of Symposium on Eye-Tracking Research and Applications (ETRA)*, 125–132. DOI: 10.1145 /1117309.1117356.

Binkley, D., Davis, M., Lawrie, D., Maletic, J.I., Morrell, C., Sharif, B. (2013). The impact of identifier style on effort and comprehension. *Empirical Software Engineering*, *18*(2), 219–276.

Busjahn, T., Shulte, C., Busjahn, A. (2011). Analysis of code reading to gain more insight in program comprehension. *Proceedings of the 11th Koli Calling International Conference on Computing Education Research*, 1–9.

Busjahn, T., Bednarik, R., Begel, A., Crosby, M., Paterson, J., Schulte, C., Sharif, B., Tamm, S. (2015). Eye movements in code reading: relaxing the linear order. *Proceedings of the IEEE 23rd International Conference on Program Comprehension*, 255–265.

Dimitri, G.M. (2015). The impact of syntax highlighting in Sonic Pi. *Proceedings of the 26th Annual Conference of the Psychology of Programming Interest Group (PPIG 2015)*, 59–68.

Futschek, G. (2006). Algorithmic Thinking: The Key for Understanding Computer Science. R.T. Mittermeir (Ed.). *Informatics education – the bridge between using and understanding computers*, Vol. 4226, 159–168. DOI: 10.1007/11915355_15.

Garner, S. (2002). Reducing the Cognitive Load on Novice Programmers. P. Barker, S. Rebelsky (Eds.). *Proceedings of ED-MEDIA 2002 World Conference on Educational Multimedia, Hypermedia & Telecommunications,* 578–583. Denver, Colorado, USA: Association for the Advancement of Computing in Education (AACE).

Gomes, A., Mendes, A.J. (2007). Learning to program – difficulties and solutions. *Proceedings of the ICEE 2007 – International Conference on Engineering Education*. Retrieved from: http://icee2007.dei.uc.pt/proceedings/papers/411.pdf.

Govender, I., Grayson, D. (2006). Learning to program and learning to teach programming: A Closer Look. E. Pearson, P. Bohman (Eds.). *Proceedings of ED-MEDIA 2006 – World Conference on Educational Multimedia, Hypermedia & Telecommunications,* 1687–1693.

Lai, M.L., Tsai, M.J., Yang, F.Y, Hsu, C.Y., Liu, T.C., Lee, S.W., Lee, M.H, Chiou, G.L., Liang, J.C, Tsai, C.C. (2013). A review of using eye-tracking technology in exploring learning from 2000 to 2012. *Educational Research Review*, 10, 90–115.

Moström, J.E. (2011). *A Study of Student Problems in Learning to Program*. Umea, Sweden: Department of Computing Science Umea University.

Ozcelik, E., Karakus, T., Kursun, E., Cagiltay, K. (2009). An eye-tracking study of how color coding affects multimedia learning. *Computers & Education*, *53*(2), 445–453. DOI: 10.1016/j.compedu.2009.03.002.

*Rozporządzenie Ministra Edukacji Narodowej z dnia 14 lutego 2017 r. w sprawie podstawy programowej wychowania przedszkolnego oraz podstawy programowej kształcenia ogólnego dla szkoły podstawowej, w tym dla uczniów z niepełnosprawnością intelektualną w stopniu umiarkowanym lub znacznym, kształcenia ogólnego dla branżowej szkoły I stopnia, kształcenia ogólnego dla szkoły specjalnej przysposabiającej do pracy oraz kształcenia ogólnego dla szkoły policealnej* (Dz.U. z 2017, poz. 356, tom 1).

Sarkar, A. (2015). The impact of syntax colouring on program comprehension. *Proceedings of the 26th Annual Conference of the Psychology of Programming Interest Group (PPIG 2015)*, 49–58.

Sharif, B., Falcone, M., Maletic, J.I. (2012). An eye-tracking study on the role of scan time in finding source code defects. *Proceedings of Symposium on Eye-Tracking Research and Applications (ETRA)*, 381–384.

Stolińska A., Andrzejewska, M. (2017a). Testing cognitive loads in solving algorithmic tasks. *15th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 1–6. DOI: 10.1109/ICETA. 2017.8102530.

Stolińska, A., Andrzejewska, M. (2017b). Analysis of the strategy used to solve algorithmic problem – a case study based on eye-tracking research. P. Dang, M. Ku, T. Qian, L.G. Rodino (Eds.). *New trends in analysis and interdisciplinary applications*, 67–76. DOI: 10.1007/978-3-319-48812-7_11.

Uwano, H., Nakamura, M., Monden, A., Matsumoto, K.I. (2006). Analyzing individual performance of source code review using reviewers' eye movement. *Proceedings of Symposium on Eye-Tracking Research and Applications (ETRA)*, 133–140.

Wing, J.M. (2006). Computational thinking. *ACM*, *49*(3), 33–35. DOI: 10.1098/rsta.2008.0118.

## Abstract

*The ability to solve problems using algorithms plays an increasingly important role in modern society, whereas programming, alongside with broadly understood digital skills, are considered to be one of the key competences of the future. The result of this trend is the modification in the Polish education system of the IT subjects' core curriculum, under which teaching programming is planned at every stage of education.*

*While teaching the skills of programming is important, it is not an easy task to achieve and hence it poses many methodological challenges. Researchers in this field are increasingly turning to new experimental methods, such as eye tracking techniques that allow to gain an insight into cognitive mechanisms and thus can provide objective information about the process of learning programming.*

*The article discusses the results of authors' own study, in which the state of declarative and procedural knowledge of students related to the forms of algorithm presentation was diagnosed. The questionnaires along with the tasks included in them, which the students solved in a traditional way and with the means of eye tracking techniques, were used in the study to track the process of solving comparable tasks presented on a computer monitor. The indicator of operational knowledge was the effectiveness of problem solving. The research was conducted on a group of 48 third-grade junior secondary school students.*

*The obtained results (low level of correct answers) indicate that the situation in the area of learning the algorithmic skills of students requires improvement. The measurement data obtained using eye tracking allowed for an in-depth analysis and interpretation of visual activity of students. Therefore, it seems that eye tracking can be considered as a complementary research technique, enriching the state of knowledge on cognitive mechanisms that are triggered in the process of solving algorithmic problems.*

**Keywords:** algorithmic problems solving, flowchart of algorithms, teaching and learning programming, eye movement parameters, eye tracking

**Magdalena Andrzejewska**, a humanities Ph.D. in the field of pedagogy holds the post of an assistant professor at the Institute of Computer Science of the Pedagogical University of Cracow. Her research interests include the issues related to learning and teaching programming, in particular the study of cognitive processes while solving algorithmic problems using eye-tracking techniques.

**Anna Stolińska** a humanities Ph.D. in the field of pedagogy holds the post of an assistant professor at the Institute of Computer Science of the Pedagogical University of Cracow. Her research interests focus on the study of human-computer interaction, with particular emphasis on cognitive processes and visual competences as well as teaching and learning computer science.