Evaluation of Eye-Movement Metrics in a Software Debugging Task using GP3 Eye Tracker

Attila Kovari[1], Jozsef Katona[1], Cristina Costescu[2]

[1]University of Dunaujvaros, CogInfoCom Based LearnAbility Research Team
Tancsics M. 1/A, 2400 Dunaujvaros, Hungary
E-mail: {kovari, katonaj}@uniduna.hu

[2]Babes-Bolyai University, Faculty of Psychology and Educational Sciences
7 Sindicatelor Street, RO-400029, Cluj-Napoca, Romania
E-mail: {cristina.costescu}@ubbcluj.ro

Abstract: Teaching different programming subjects is an increasing challenge nowadays as because of growing user demands, the latest paradigms and technologies must be taught. Students in higher education now are of the age who were born into the digital world; however, the success of fulfilling programming courses is lagging behind. Human-computer interface-based research has emerged in numerous fields of science recently, which could lead to the revolutionising of education. These interfaces could also help professors as a support system in transferring knowledge in a more efficient way besides supporting students acquiring adequate home learning methods. In this study, the applicability of eye movement tracking systems in respect of a programming task is examined, in which during the exploration and correction of the errors of an incorrectly functioning algorithm, the eye movement parameters are observed, recorded and evaluated. The test subjects participating in the research were divided into two groups according to some of their characteristic parameters, where the first group during debugging rather applied minor modifications and the more common technique of compile and run, which otherwise is also the most characteristic feature of students studying programming, while the members of the other group increased emphasis on analysing. In the statistic evaluation of the research, the parameters characteristic of the eye movement tracking of the two groups, as well as the efficiency of these groups were analysed. Based on the results, regarding the efficiency and the number of fixations, a significant difference could be shown between the two groups ($U=22.5$, $Z=-2.236$, $p=0.025$ (2-tailed), $r=0.48$) and ($t(20)=2.507$, $p=0.021$ (2-tailed), $d=1.106$), while concerning the duration of fixation and the saccade length, the difference shown was infinitesimal ($t(20)=0.544$, $p=0.592$ (2-tailed), $d=0.26$) and ($t(19.992)=1.965$, $p=0.063$ (2-tailed), $d=0.79$).

Keywords: eye-tracking; programming; debugging; education

1 Introduction

Understanding complex cognitive processes can provide great help in education. Human-computer interfaces are such technologies that could emerge as a support system in the fields of teaching and learning. Teachers could achieve more efficient knowledge transfer, while students could gain adequate learning methods by applying them. Teaching programming has become a complex task nowadays, as the software solutions to be produced and maintained are growing in size and they are increasingly difficult to handle despite the available newer and newer paradigms and technologies compared to the past; moreover, students find it hard to acquire applying these devices because of their complexity. Designing, preparing, testing and maintaining source code bases are becoming a more complex cognitive process; therefore, research aiming to increase the efficiency of development phases and reduce their costs are being in focus. One of the possibilities is the analysis and examination of eye movement tracking, whereas the observation of gaze can provide an opportunity to explore basic and more complex cognitive mechanisms [1].

The [1] research deals with the metrics of the route of the gaze in respect of software development; it encourages the standardisation of these features and makes various suggestions for their applications. Some studies examine the steps of the designing phase of the software development life cycle in which UML class diagrams that create the frame of object-oriented programs and are strongly present in higher education are designed and applied [2-4], moreover, further studies analyses the arrangement [5, 6] and intelligibility of these diagrams [7]. The understand ability of Business Process Models (BPMN diagrams) is examined in the study [8], while that of the Entity-Relationship Diagrams (ERD) is examined in [9]. The reading of traditional, native language texts is analysed and compared to the readability of different source codes in other articles. The results among others showed that the methods of the reading and overview of the two text types are different, as in case of program codes, more fixation time could be shown. Another result proves that beginner programmers spend more time reading comments than experienced programmers [10-13]. The study [14] was the first research in which large source code was tested in relation to eye movement with open source code software support.

In this research, the forms and efficiency of debugging sections of software development are examined with eye movement tracking with the involvement of test subjects. The route of the gaze of the test subjects was observed continuously during the process, and coherences were determined by the evaluation of the recorded metrics. The eye movement tracking hardware and software units, as well as the test algorithm applied in the research are described in Chapter 2.The circumstances of the inquiry are introduced in Chapter 3, including the data of the test subjects and the steps of the completion of the research. The evaluation of the results is detailed in Chapter 4, while in Chapter 5, the conclusions made from the results are summarised.

2 The Hardware Unit, the Software Package and the Algorithm Applied in the Research

The GazePoint 3 (GP3) eye-tracker hardware unit and the OGAMA software package were used in the research to observe eye-movement tracking and to record the metrics. The applied algorithm of the examination, uniting two disordered blocks (unification), was well known for the test subjects from their previous studies.

2.1 The GazepointGP3 Eye-Tracker Hardware Unit

During the study, to observe and record the route of the gaze of the test subjects, a general-purpose research-grade device, the GazepointGP3 eye-tracker (Fig. 1) was applied, which had successfully been applied in some previous research. [15-21] It is an ultra-portable device (320 x 45 x 40 mm, 145 g) that can move 25 cm horizontally, 11 cm vertically and 15 cm in-depth; it can be fitted on the monitor and uses infra camera observation and image procession to detect and follow eye movement with 60 Hz sample rate. It is easy to handle with 0.5-1 degree of visual angle and 5-or 9 point-calibration options. The device can be used at least in case of 24” or smaller displays, with at least Intel Core i7 or faster processors, with 8 GB RAM and with Windows 7, 8.1 or 10 operating system. At present, the Mac and Linux operating systems are not supported. In addition, it also has an API/SDK package that supports software development possibilities related to the device.

Fig. 1.
The GazepointGP3 Eye Tracker

2.2 The OGAMA Software Package

The OGAMA (OpenGazeAndMouseAnalyzer) is an open-source code application that was implemented by Adrian Voßkühler at the Freie Universität Berlin to track eye and mouse movements and to record and analyse the received parameters implemented in C# high-level programming language.NET frame system. For the adequate use of the application, an eye movement tracking hardware unit, a Windows operating system by Microsoft, the .NET frame system and the SQL

Express database server are needed. The main features of the application include slideshow design, database-driven pre-processing, attention map creation, filtering, and recording of gaze and mouse movement data and the areas of interest definition. Furthermore, data stored in the database can be exported for different statistic software in proper formats, which eases efficient statistic evaluation. The application supports several gaze routes observing and recording hardware units, including the GazepointGP3 hardware unit as well. In addition, the program can be modified or further developed according to our needs. The software package has been successfully applied in several researches. [21-23]

2.3 The Applied Test Algorithm

The test subject had to correct errors hidden in unification known from their previous studies, wherein the correct original algorithm an N and an M element sets are available depicted in *A* and *B* vectors. The unification of the two sets is represented in the *C* vector, to which those elements belong to that are present in at least one of them. During the procession, the content of the *A* vector is copied into *C* at first, which has an *N* element, and then each element of the *B* vector has to be found an *A* correspondent. If such an element cannot be found, so *i>M*, the j element of the *B* vector goes to the *C* vector. The description of the original union of two unsorted array pseudo can be seen in Fig. 2.

ALGORITHM ORIGINAL UNION OF TWO UNSORTED ARRAYS (*A, B, C, N, M*)

```
for i ← 1 to N do
    C[i] ← A[i]
end for
count ← N
for j ← 1 to M do
    i ← 1
    while i<=N and A[i] ≠B[j]do
        i ← i + 1
    end while
    if i>=N then
        C[count] ← B[j]
        count ← count + 1
    end if
end for
```

Fig. 2.
Original Union of Two Unsorted Array Algorithm

In the correct union algorithm above, three errors were hidden, which had to be found and corrected by the test subjects. The artificially generated errors were the following:

1. The conditional statement part of the *for* cycle was modified to *N-1*, so the last element of the *A* vector can never be transferred into the *C* vector.

2. The index reference of the *C* vector was changed from *count* to *i*, and as a result, the last element of the *B* vector that cannot be found in the *A* vector will only be transferred to the *N* (*A.length+1*) position of the *C* vector.

3. Within the *if* conditional branch, the incremental expression (*count* = *count* + *1*) was changed into a decremental expression (*count* = *count* - 1) and a result of this, the first element of the *B* vector that cannot be found in the *A* vector will only be transferred to the *N* (*A.length+1*) position of the *C* vector.

The description of the modified union of two unsorted array pseudo can be seen in Fig. 3, where the modifications are indicated in red.

ALGORITHM MODIFIED UNION OF TWO UNSORTED ARRAYS (*A, B, C, N, M*)

```
for i ← 1 to N-1 do
    C[i] ← A[i]
end for
count ← N
for j ← 1 to M do
    i ← N
    while i<= N and A[i] ≠ B[j] do
        i ← i + 1
    end while
    if i>=N then
        C[i] ← B[j]
        count ← count - 1
    end if
end for
```

Fig. 3.
Modified Union of Two Unsorted Array Algorithm

The modified algorithm demonstrated above was implemented in C# programming language using the Visual Studio Community development environment, which was best known for the test subjects.

3 The Circumstances of the Examination

The research was carried out with the application of the Gazepoint GP3 eye-tracker hardware unit, the OGAMA software package and the modified unification algorithm all described in detail in the previous chapter. The test subjects were given a maximum of three minutes to explore and correct the hidden errors, and they were allowed to compile and run the application several times. An LG 22M45 type with 1920x1080 resolution, 22” diameter monitor was applied to display the modified algorithm.

3.1 The Test Subjects

Twenty-two university students, ten females and twelve males aged between 18 and 22, who claimed themselves to be wholly healthy and were not on any medication, were involved voluntarily in the research. The only condition of the participation was the successful compliance of the Introduction to Programming course, as this subject contains the necessary algorithm knowledge and algorithmic thinking to complete the test.

3.2 The Steps of the Test

Before connecting the GP3 eye-tracker hardware unit, the Gazepoint software package was installed, which contains the camera driver among others. After successful installation, the device was connected through a USB port. After connecting the device, it was positioned under the monitor approximately 65 cm length distance from the eyes. It was avoided in every case that the face of the test subjects is exposed to sunlight. After the adequate placement of the device, the modified algorithm was opened in the Visual Studio Community development environment. It was positioned to fill the screen as much as it could. After opening the modified algorithm, the Gazepoint Control application was started, which supported the configuration and the start of the gaze-date server, ensuring real-time information obtaining. The next step was to start the OGAMA software, where before beginning the test, the Recording module was selected in the OGAMA application and the connection to the hardware unit was completed. After the preparations, the test subjects were seated one after another, one by one, and the testing process was explained to them. Right before starting the test, the data of the test subjects were saved. After successful data registration, the calibration of the device had to be done, during which the test subjects had to track a circle with eye movement from the top left corner of the monitor without moving their heads.

For the best results, calibration was done even more times in the case of each person. After successful calibration, the test subjects could start the error search and correction. During the research, the different eye movement parameters were observed and recorded, and after finishing the test, the data were saved into a database for further statistic evaluation. The process and the environment of the test can be seen in Fig. 4.

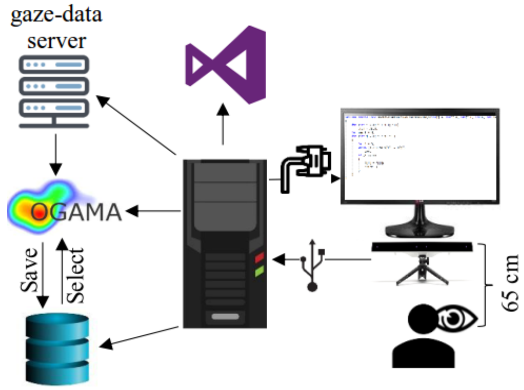

Fig. 4.
A schematic diagram of the equipment setup

3.3 Definied Metrics

The successfulness was defined on a scale between 0 and 3, where the points of efficiency were equal to the numbers of the found bugs. The test subjects had a maximum of 3 minutes to detect and correct the errors because the time factor would be used as a measure of efficiency during the examination.

4 Results

During the evaluation of the recorded results, two groups were set based on the way of problem-solving. Test subjects who after many minor modifications tried to explore and correct the errors with more compile and run were put into the first group (Try&R), while those who rather analysed the algorithm and applied compile and run relatively fewer times, maximum at five times, were ranked into the second group (Think&R).

4.1 Testing the Efficiency and the Rapidity of the Groups

It became clear after the test that more, actually 15 test subjects, applied smaller corrections and multiple runs, so they rather “tried” to do their task. Therefore, they were put in the Try&R group, while 7 test subjects showed more consideration and analysing, so they rather “thought” and were placed therefore in the Think&R group.

Based on the Try&R results, it can be claimed that two people in the group were able to absolve the task, despite many smaller modifications and attempts; however, two other test subjects were not able to correct any errors. Regarding time, it took 84 seconds to find and correct all errors for the quickest test subject, while thirteen test subjects reached the maximum time limit. The average of efficiency regarding the whole group was 1.67±0.90. More test subjects ran out of the given three minutes, presumably because of their precipitance and little consideration. The average time was 169.067±29,33 seconds regarding the whole group. All in all, two of the fifteen test subjects were able to find and correct all errors within the target time, and it shows 13% successfulness. Fig. 5 shows the distribution of efficiency of the Try&R group, which greatly shows that two errors were found and corrected within the maximum time limit by most test subjects.

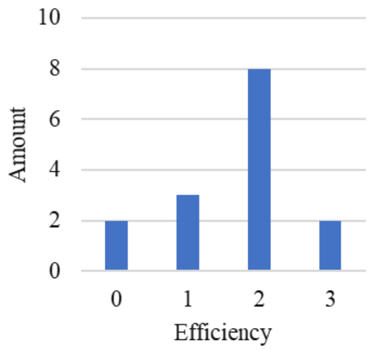

Fig. 5.
The distribution of efficiency of the Try&Rgroup

According to the results of the Think&R group, it can be stated that five people were able to solve the task correctly; however, there were not any test subjects who could not find and correct any errors, the worst result was finding and correcting only one error. Regarding the time, the fastest test subject needed a little bit more than one minute to find and correct all mistakes, while three people reached the maximum time limit. The average efficiency of the whole group was 2.57±0.79. Two test subjects ran out of the three minutes, but proportionally they were much less than in the previous group. The average time of solving the task was 136.86±44.13 seconds regarding the whole group. Overall, 5 out of the 7 people were able to find and correct all errors within the target time, and that shows 71% successfulness, which means significant positive difference comparing to the results of the Try&R group. Presumably more careful consideration and less precipitance lie behind the better results. Fig. 6 shows the distribution of efficiency of the Think&R group, which reflects that most test subjects, could find and correct all errors within the given time limit.

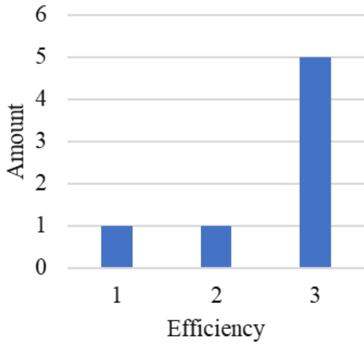

Fig. 6.
The distribution of efficiency of the Think&R group

A summary of the evaluation above can be seen in Table I, showing the efficiency of each test subject and the time necessary to fulfil the task. At the bottom of the table, the minimum and maximum value of the results reached in the groups, their average, and standard deviation are given.

TABLE I.
A SUMMARY TABLE ABOUT THE EFFICIENCY OF THE GROUPS

	TRY&R		THINK&R	
Test subjects	**Efficiency[1] (points)**	**Time[2] (s)**	**Efficiency[1] (points)**	**Time[2] (s)**
1.	1	180	3	134
2.	1	180	3	172
3.	2	180	2	180
4.	0	180	1	180
5.	2	180	3	124
6.	1	180	3	62
7.	3	112	3	106
8.	2	180		
9.	2	180		
10.	2	180		
11.	0	180		
12.	2	180		
13.	3	84		
14.	2	180		
15.	2	180		
Min	**0**	**84**	**1**	**62**
Max	**3**	**180**	**3**	**180**
Average	**1.67±0.90**	**169.07±29.33**	**2.57±0.79**	**136.86±44.13**

[1]The maximum points were 3.
[2]The maximum time was 180 s.

The data of average efficiency in case of the Try&R and the Think&R groups were analysed on an ordinal scale, therefore, it was determined with the Mann-Whitney test that between the average of the efficiency of the two groups, a statistically significant difference could be found (U=22.5, Z=-2.236, p=0.025 (2-tailed), r=0.48) taking into account the maximum 180 seconds time limit.

4.2 The Evaluation of the Parameters of the Eye Movement

During the examination of the parameters of the eye movement, the attention maps generated by the OGAMA software were analysed at first, which were determined while recording the route of the gaze. Besides the attention map, the number of fixations, the average fixation duration as well as the average saccade lengths in pixels was examined, too.

4.2.1 The Evaluation of the Recorded Attention Maps

The applied colours in the attention map mean the following:

- transparent field: observed, focused area for only a very short time or not at all;
- green: observed, focused area for a short time;
- yellow: observed, focused area for a medium length of time;
- red: observed, focused area for a longer time.

Fig. 7 shows a map characteristic of the Try&R group, while Fig. 8 shows that of the Think&R group. Regarding the members of the two groups, the received results are similar. In case of the Try&R group, it can be seen that the route of gaze is more diversified; the less important part of the algorithm was paid more attention than necessary, too. It reflects the hesitancy well, as presumably, the gaze returned to a previously relatively shortly observed and examined field, even if that certain code part did not contain any errors and the test subjects could have been ascertained about that earlier. On the whole, it can be stated that a characteristic feature of the members of the Try&R group is diverse attention, which is also proved by the generated attention maps.

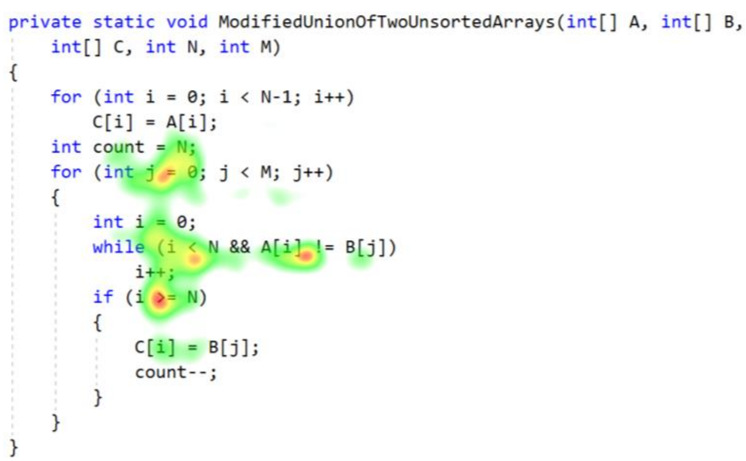

Fig. 7.
The attention map of the 12th test subject in the Try&R group (Efficiency = 2, Time = 180 s)

In case of the Think&R group, it could be observed that the less important areas were paid little attention to according to the gaze route, while the incorrect parts of the algorithm were more focused on than in case of the members of the Try&R group. It can assume more careful consideration and concentrated attention, which is proven by the attention maps.

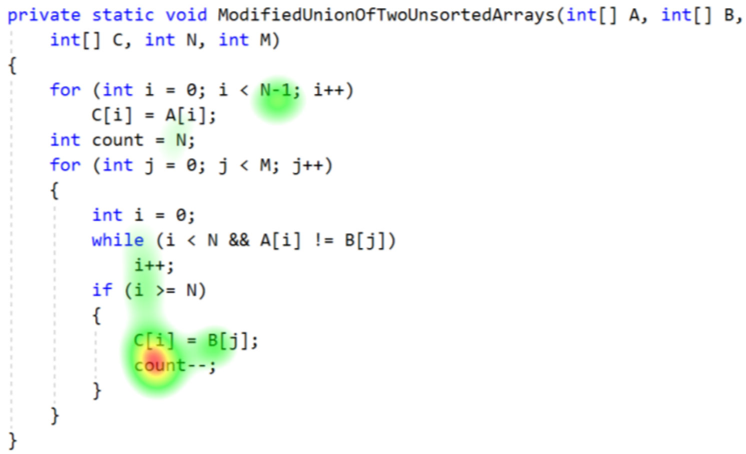

Fig. 8.
The attention map of the 5th test subject in the Think&Rgroup (Efficiency = 3, Time = 124s).

4.2.1 The Evaluation of the Index Numbers of the Eye Movement

The first evaluation of the eye movement index number was defined by the number of fixations. In case of the Try&R and the Think&R groups, during the examination of normality of the fixation data, the Shapiro-Wilk (D(15)=0.904, p=0.110 and D(7)=0.901, p=0.339) test results are not significant, and the standard deviations are not consentaneous either (F=0.021, p=0.887), furthermore, at the evaluation, the average of interval variables were compared in two groups independent of each other, therefore, two-sample t-test was applied, which shows that the average of the number of fixations is significantly different in the two groups ($t(20)=2.507$, $p=0.021$ (2-tailed), d=1.106). In case of the Try&R group, the average fixation quantity is 210.67±50.06, while in the Think&R group, it is 149.57±59.98. The distribution of the number of fixations is shown in Fig. 9.

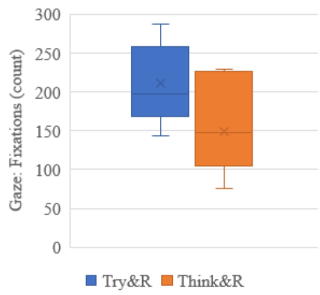

Fig. 9.
The distribution of Gaze: Fixations (count) of the groups

The second evaluation of the eye movement index number was carried out regarding the average fixation duration. In case of the Try&R and the Think&R groups, during the examination of normality of the average fixation duration data, the Shapiro-Wilk test results ($D(15)=0.960$, $p=0.691$ and $D(7)=0.944$, $p=0.672$) are not significant, and the standard deviations are not consentaneous either (F=0.654, p=0.428), therefore, as in case of the evaluation of the quantity of fixation, the two-sample t-test was applied, which shows that the average of the fixation period of time is not significantly different in the two groups ($t(20)=-0.544$, $p=0.592$ (2-tailed), $d=0.26$). In case of the Try&R group, the average fixation period is 364.01±85.62, while in the Think&R group, it is 383.96±65.71. The distribution of the average fixation duration is shown in Fig. 10.

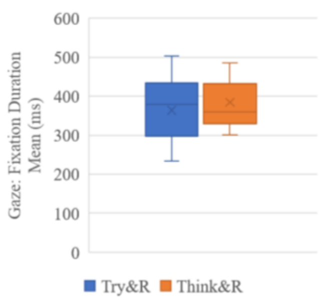

Fig. 10.
The distribution of Gaze: Fixation Duration Mean (ms) of the groups

The third and last evaluation of the eye movement index number was executed regarding the average saccade length. In case of the Try&R and the Think&R groups, during the examination of normality of the average saccade length, the Shapiro-Wilk test results ($D(15)=0.932$, $p=0.294$ and $D(7)=0.838$, $p=0.096$) are not significant, but the standard deviations are consentaneous either (F=7.333, p=0.014), the two-sample t-test (Welch) was applied, which shows that the average of the saccade length is not significantly different in the two groups

($t(19.992)=1.965$, $p=0.063$ (2-tailed), $d=0.79$). In case of the Try&R group, the average saccade length is 125.24±29.93 pixels, while in the Think&R group, it is 106.96±13.68 pixels. The distribution of the average saccade length is shown in Fig. 11.

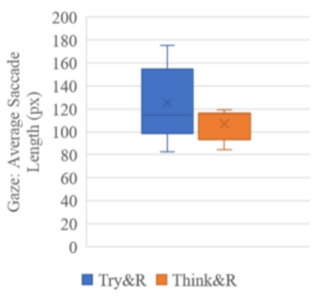

Fig. 11.
The distribution of Gaze: Fixations (count) of the groups

The summary of the evaluation of the data above can be seen in Table II, in which the quantity of fixation, the duration of average fixation and the average saccade length in pixels of each test subjects can be seen. At the bottom of the table, the minimal and the maximal values of the received results of the groups as well as their average value and standard deviations are shown.

TABLE II.
SUMMARY TABLE OF THE PARAMETERS OF THE EYE MOVEMENTSOF THE TWO GROUPS

	TRY&R			THINK&R		
Test subjects	**Gaze: Fixations (count)**	**Gaze: Fixation Duration Mean (ms)**	**Gaze: Average Saccade Length (px)**	**Gaze: Fixations (count)**	**Gaze: Fixation Duration Mean (ms)**	**Gaze: Average Saccade Length (px)**
1.	144	244,48	107,49	227	329,95	92,93
2.	260	317,02	82,33	230	351,56	84,31
3.	258	352,68	95,39	148	432,01	116,40
4.	283	233,70	154,68	105	428,90	115,49
5.	189	297,55	127,47	110	360,04	116,46
6.	151	495,38	175,58	75	300,73	119,27
7.	201	311,55	94,59	152	484,54	103,88
8.	168	386,69	100,95			
9.	197	401,54	114,29			
10.	257	388,26	166,46			
11.	176	451,82	159,43			
12.	288	433,58	137,69			
13.	184	263,58	114,67			
14.	155	379,58	98,24			
15.	249	502,58	149,27			
Min	**144**	**233.70**	**82.33**	**75**	**300.73**	**84.31**
Max	**288**	**502.58**	**175.58**	**230**	**484.54**	**119.27**
Average	**210.66±50.06**	**364.01±85.62**	**125.24±29.93**	**149.57±59.98**	**383.96±65.71**	**106.96±13.68**

5 Discussion

Having examined and evaluated the results, it can be stated that in case of the test subjects of the Try&R group, the several minor modifications and the frequent compile and run shows less efficiency and larger time need to explore and correct the hidden errors in the source code than in the case of the test subjects of the Think&R group. The presumable reason for this phenomenon is uncertainty, diverse attention and the lack of consideration.

The significant difference of the evaluated eye movement parameters experienced in the results of the two groups is also confirmed by the attention maps recorded by the OGAMA software. Based on the map, in case of the Try&R group, the gaze of the test subjects is much more diverse; moreover, the test subjects of this group paid too much attention to the less important parts of the algorithm, which also proves the statement claimed above.

According to the eye movement parameters, it can be stated that a significant difference between the test subjects of the two groups could only be shown in the quantity of fixation. The test subjects in the Try&R group performed more fixations in the source code compared to the members of the Think&R group, that is they needed to gain information from more points of the screen, however, regarding the duration of fixation and the average saccade length, there was no significant difference ivincible, which may refer to the fact that despite the members of the Try&R group examined more points in the source code, they did not study them thoroughly, furthermore, during the visual search many small scanning characterised them, while in the Think&R group these results may mean that longer or shorter fixation duration or visual search was not necessary besides smaller quantity of fixation, so the better results received in the Think&R group is the consequence of the fact that the test subjects needed significantly less fixation to find and correct the errors. As in the duration of the information process, a significant difference cannot be shown, and all test subjects had successfully absolved the Introduction to Programming course, the difference cannot be explained with the knowledge difference of the test subjects. The knowledge gap presumably would have shown larger information processing duration and average saccade length besides the larger quantity of fixation because of indecision.

The findings of the examination suggest that paying attention and consideration to the problems could be a better way of doing the task, leading to efficient debugging. These results can useful for teachers in software development when discussing the best and efficient ways of debugging. In addition, this kind of researches can be of interest not only to science and education, but also to industry. Using the typical patterns and metrics the comprehensibility of a source code can be effectively analysed and the cost of its future maintenance can be estimated with more accurately. Moreover, bug fixes and testing may be accelerated, so the complete development time could be shortened since developers and testers who can understand the whole source code may be discovered. In the future, different software development companies may invest in human-machine interfaces that can provide them more efficient development.

Conclusions

The research on eye movement tracking is increasingly present in such examinations of cognitive processes as programming. This research was carried out by observing the route of the gaze of test subjects during the exploration and correction of an incorrect algorithm and the recorded eye movement parameters were analysed. After the experiment, two groups were formed by applying parameters characteristic of the test subjects, where the first group (Try&R) rather applied the technique of smaller modifications and more frequent compiling and run during debugging, while the members of the second group (Think&R) put more emphasis on analysing and applied less compiling and run of the application. Besides analysing the efficiency, the heat maps generated by the software applied for eye movement tracking were analysed as well, in which the route of the gaze of each test subjects could be tracked well. During the statistic evaluation, the parameters characteristic of the eye movement tracking of the two groups (quantity of fixation, average fixation duration, and average saccade length) and the efficiency of the groups were analysed. As aresult of the research, it can be stated that in case of less precipitance, thorough consideration and attention, less information procession quantity is sufficient to efficient debugging, which can also lead to more efficient software development.

The evaluation of the eye movement parameters could serve as a support for measuring different abilities: from cognitive skills, learning strategies to complex task execution in the field of education. The modern ICT possibilities help achieve these findings using the advantages of VR and AR [24]-[28], mathability [29]-[34], gamification, project-based learning or cooperative methodologies [35]-[41] together with the possibilities of Cognitive Info-communications and similar emerging technologies [42]-[48].

Acknowledgement

This research was supported by EFOP-3.6.1-16-2016-00003 grants, Establishment long-term R & D & I process at the University of Dunaújváros.

References

[1] Z. Sharafi, T. Shaffer, B. Sharif and Y. Guéhéneuc: Eye-Tracking Metrics in Software Engineering, *2015 Asia-Pacific Software Engineering Conference (APSEC)*, New Delhi, 2015, pp. 96-103

[2] B. De Smet, L. Lempereur, Z. Sharafi, Y. Guéhéneuc, G. Antoniol and N. Habra: Taupe: Visualizing and analyzing eye-tracking data, Science of Computer Programming, Vol. 79, 2014, pp. 260-278

[3] G. Cepeda Porras and Y. Guéhéneuc: An empirical study on the efficiency of different design pattern representations in UML class diagrams, Empirical Software Engineering, Vol. 15, No. 5, 2010, pp. 493-522

[4] S. Jeanmart, Y.-G. Guéhéneue, H. A. Sahraoui, N. Habra: Impact of the visitor pattern on program comprehension and maintenance, *Proceedings of 3rd International Symposium on Empirical Software Engineering and Measurement*, 2009, pp. 69-78, 2009

[5] S. Yusuf, H. H. Kagdi, J. I. Malefic: Assessing the comprehension of UML class diagrams via eye-tracking, *Proceeding of 15th IEEE International Conference on Program Comprehension ser. ICPC '07*, 2007, pp. 113-122

[6] B. Sharif, J. I. Maletic: An eye-tracking study on the effects of layout in understanding the role of design patterns, *Proceedings of the 26th IEEE International Conference on Software Maintenance*, 2010, pp. 1-10

[7] Z. Soh, Z. Sharafi, B. V. den Plas, G. C. Porras, Y.-G. Guéhéneuc, G. Antoniol: Professional status and expertise for UML class diagram comprehension: An empirical study, 2012, pp. 163-172

[8] R. Petrusel, J. Mendling: Eye-tracking the factors of process model comprehension tasks, *Proceedings of the Conference on the Advanced Information Systems Engineering*, 2012, pp. 224-239

[9] N. E. Cagiltay, G. Tokdemir, O. Kilic, D. Topalli: Performing and analyzing non-formal inspections of entity relationship diagram (erd), *Journal of Systems and Software*, Vol. 86, No. 8, 2013, pp. 2184-2195

[10] M. E. Crosby, J. Stelovsky: How do we read algorithms? a case study", Vol. 23, No. 1, 1990, pp. 24-35

[11] M. E. Crosby, J. Scholtz, S. Wiedenbeck: The roles beacons play in comprehension for novice and expert programmers, *Proceeding of Programmers 14th Workshop of the Psychology of Programming Interest Group*, 2002, pp. 18-21

[12] T. Busjahn, C. Schulte, A. Busjahn: Analysis of code reading to gain more insight in program comprehension, *Proceedings of the 11th Koli Calling International Conference on Computing Education Research ser. Koli Calling' 11*, 2011, pp. 1-9

[13] T. Busjahn, R. Bednarik, C. Schulte: What influences dwell time during source code reading?: Analysis of element type and frequency as factors, *Proceedings of the Symposium on Eye Tracking Research & Applications ser. ETRA '14. New York*, 2014, pp. 335-338

[14] K. Kevic, B. M. Walters, T. R. Shaffer, B. Sharif, T. Fritz, D. C. Shepherd: Tracing software developers eyes and interactions for change tasks, *Proceedings of the 10th Joint Meeting of the European Software*

Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, 2015, pp. 202-213

[15] S. Seha, G. Papangelakis, D. Hatzinakos, A. S. Zandi and F. J. Comeau: Improving Eye Movement Biometrics Using Remote Registration of Eye Blinking Patterns, *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, 2019, pp. 2562-2566

[16] H. Zhu, S. Salcudean and R. Rohling: A novel gaze-supported multimodal human–computer interaction for ultrasound machines, International Journal of Computer Assisted Radiology and Surgery, 2019, pp. 1-9

[17] J. Meng, T. Streitz, N. Gulachek, D. Suma and B. He: Three-Dimensional Brain–Computer Interface Control Through Simultaneous Overt Spatial Attentional and Motor Imagery Tasks, in *IEEE Transactions on Biomedical Engineering*, Vol. 65, No. 11, 2018, pp. 2417-2427

[18] J. Iskander*et al.*: Age-Related Effects of Multi-screen Setup on Task Performance and Eye Movement Characteristics," *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Miyazaki, Japan, 2018, pp. 3480-3485

[19] B. J. Edelman, J. Meng, N. Gulachek, C. C. Cline and B. He: Exploring Cognitive Flexibility With a Noninvasive BCI Using Simultaneous Steady-State Visual Evoked Potentials and Sensorimotor Rhythms, in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 26, No. 5, 2018, pp. 936-947

[20] S. Li and X. Zhang: Implicit Intention Communication in Human–Robot Interaction Through Visual Behavior Studies, in *IEEE Transactions on Human-Machine Systems*, Vol. 47, No. 4, 2017, pp. 437-448

[21] A. Voßkühler, V. Nordmeier, L. Kuchinke and A. Jacobs: OGAMA (Open Gaze and Mouse Analyzer): Open-source software designed to analyze eye and mouse movements in slideshow study designs, Behavior Research Methods, Vol. 40, No. 4, 2008, pp. 1150-1162

[22] T. Ujbányi: Examination of eye-hand coordination using computer mouse and hand tracking cursor control, *2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, Budapest, Hungary, 2018, pp. 353-354

[23] T. Ujbanyi, J. Katona, G. Sziladi and A. Kovari: Eye-tracking analysis of computer networks exam question besides different skilled groups, *2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, Wroclaw, 2016, pp. 277-282

[24] Ildikó Horváth: Evolution of teaching roles and tasks in VR/AR-based education, Proceedings of the 9th IEEE International Conference on

Cognitive Infocommunications (CogInfoCom), Budapest, 2018, pp. 355-360

[25] Tibor Guzsvinecz, Csaba Kovacs, Dominich Reich, Veronika Szucs, Cecilia Sik-Lanyi: Developing a virtual reality application for the improvement of depth perception, 9th IEEE International Conference on Cognitive Infocommunications, 2018, pp. 17-22

[26] Cecilia Sik-Lányi,Veronika Szücs, Janet Stark: Virtual reality environments development for aphasic clients, International Journal of Stroke, 2014, pp 241-250

[27] Adam Csapo, Ildiko Horvath, Peter Galambos and Peter Baranyi: VR as a Medium of Communication: from Memory Palaces to Comprehensive Memory Management, Proceedings of the 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Budapest, 2018, pp. 389-394

[28] Ildiko Horvath: Innovative engineering education in the cooperative VR environment, Proceedings of the 7th IEEE Conference on Cognitive Infocommunications (CogInfoCom), Wrocław, 2016, pp. 359-364

[29] Peter Baranyi, Attila Gilányi: Mathability: Emulating and enhancing human mathematical capabilities, Proceedings of the 4th International Conference on Cognitive Infocommunications (CogInfoCom), Budapest, 2013, pp. 555-558

[30] Katarzyna Chmielewska, Attila Gilányi: Mathability and computer aided mathematical education, Proceedings of the 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Gyor, 2015, pp. 473-477

[31] Katarzyna Chmielewska, Attila Gilányi, A. Łukasiewicz: Mathability and mathematical cognition, Proceedings of the 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Wroclaw, 2016, pp. 245-250

[32] Katarzyna Chmielewska, Attila Gilányi: Computer Assisted Activating Methods in Education, Proceedings of the 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Naples, 2019, in press

[33] Gilányi A.: Charakterisierung von monomialen Funktionen und Lösung von Funktionalgleichungen mit Computern, PhD Thesis, University of Karlsruhe, 1995

[34] Gilányi A.: Solving linear functional equations with computer, Math. Pannon., 9:57–70, 1998

[35] Ilona Heldal, Carsten Helgesen: The Digital HealthLab: Supporting Interdisciplinary Projects in Engineering and in Health Education, Journal

of Applied Technical and Educational Sciences, Vol. 8, No. 4, 2018, pp. 4-21

[36] György Molnár: The Impact of Modern ICT-based Teaching and Learning Methods in Social Media and Networked Environment, Proceedings of the 11th International Scientific Conference on Distance Learning in Applied Informatics, Nitra, 2016, pp. 341-350

[37] Robert Pinter, Sanja Maravic Cisar: Measuring Team Member Performance in Project Based Learning, Journal of Applied Technical and Educational Sciences, Vol. 8, No. 4, 2018, pp. 22-34

[38] Laszlo Bognar, Eva Fancsikne, Peter Horvath, Antal Joos, Balint Nagy, Gyorgyi Strauber: Improved learning environment for calculus courses, Journal of Applied Technical and Educational Sciences, Vol. 8, No. 4, 2018, pp. 35-43

[39] György Molnár: The role of electronic and virtual learning support systems in the learning process, Proceedings of the 8th International Symposium on Applied Computational Intelligence and Informatics, Timisoara, 2013, pp. 51-54

[40] Igor Fürstner, Laszlo Gogolák, Peter Sarcevic: Development of telepresence technology during the teaching process at Subotica Tech, Journal of Applied Technical and Educational Sciences, Vol. 8, No. 4, 2018, pp. 44-53

[41] E. Gogh, A. Kovari, Experiences of Self-regulated Learning in a Vocational Secondary School, Journal of Applied Technical and Educational Sciences, Vol. 9, No. 2, 2019, pp. 72-86

[42] Attila Kovari: CogInfoCom Supported Education: A review of CogInfoCom based conference papers, Proceedings of the 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2018, pp. 233-236

[43] Peter Baranyi, Adam Csapo, Gyula Sallai, Cognitive Infocommunications (CogInfoCom), Springer International Publishing Switzerland, 2015, 219 p

[44] György Molnár, András Benedek: ICT Related Tasks and Challenges In The New Model of Technical Teacher Training, Proceedings of the InfoWare 2013 - ICCGI 2013 the Eighth International Multi-Conference on Computing in the Global Information Technology, Nice, 2013, pp. 40-44

[45] Imre Petkovics: Digital Transformation in Higher Education, Journal of Applied Technical and Educational Sciences, Vol. 8, No. 4, 2018, pp. 77-89

[46] Tibor Guzsvinecz, Veronika Szucs, Cecilia Sik-Lanyi: Suitability of the Kinect Sensor and Leap Motion Controller - A Literature Review SENSORS, Vol. 19, No. 5, 2019, pp. 1072-1097

[47] Veronika Szucs, Tibor Guzsvinecz, Attila Magyar: Improved algorithms for movement pattern recognition and classification in physical rehabilitation, 10th IEEE International Conference on Cognitive Infocommunications, 2019, pp. 417-424

[48] Veronika Szucs, Gyorgy Karolyi, Amdras Tatar, Attila Magyar: Voice Controlled Humanoid Robot based Movement Rehabilitation Framework, 9th IEEE International Conference on Cognitive Infocommunications, 2018, pp. 191-196.