

Analyse the Readability of LINQ Code using an Eye-Tracking-based Evaluation

Jozsef Katona

University of Dunaújvaros, Informatics Institute, Department of Software
Development and Application
CogInfoCom Based LearnAbility Research Team
Táncsics M. 1/A, 2400 Dunaújvaros, Hungary
E-mail: katonaj@uniduna.hu

Abstract: Nowadays, in addition to traditional procedures, the use of eye movement tracking systems can be an alternative possibility, to analyse the effectiveness of different programming technologies. Tracking the gaze route by analysing eye movement parameters also allows the study of a complex cognitive process. In this research an eye-tracking system is presented besides the traditional knowledge level test to analyse the readability of the two semantically identical but syntactically different options of the Language-Integrated-Query (LINQ) abstraction layer, the query and the method syntax. Each query operator and extension method were compared besides the complete query expressions in different syntaxes. The observed and recorded results are evaluated with descriptive statistics. Based on the obtained results, it can be stated that the application of eye movement tracking systems, in the study of complex cognitive processes, such as, programming, is well applicable and can also provide additional information for teachers, developers or researchers concerning adequate syntax, to present or write more efficient, maintainable and further developable source code.

Keywords: eye-tracking; programming; comprehension; cognition load; LINQ

1 Introduction

The eye-tracking systems measures the direction of the eye gaze and records the movement of the eye, relative to the head. These systems are being used successfully in more and more fields of science. General eye-hand coordination task in relation to mouse cursor movement on computer screen, can be examined and the efficiency of computer mouse and gesture-based cursor position control from the perspective of eye-hand coordination, can be also compared. Research [1] [2] concentrates on multi-criteria decision analysis of features of recommending interfaces, taking into account their visual intensity, attention represented by fixations measured, with eye tracking and time required to attract

attention after a website is loaded. Paper [3] proposes a validated framework for the performance evaluation of a recommending interface, to optimize its efficiency considering individual user characteristics. Study [4] uses a quantitative research methodology involving gaze tracking for implicit monitoring of human-website interaction in an experiment instrumented for a simple structure recommending interface.

Increasingly complex software systems set increasing challenges to software developers. Writing and maintaining source code would become virtually impossible without the development of programming technology. In addition to newer and newer programming capabilities, observing, examining and evaluating eye movement parameters could cause changes and improvements in the reading, interpretation, and debugging mechanisms of the source code by the developers, ultimately resulting in easy-to-read and maintainable source codes.

Even in cognitive psychology, it has been observed that texts without spaces written with the camelCase technique were more difficult for test subjects to read than those written with under_score. [5] [6] The results of the research can also be related to programming, as different forms are used to name different programming elements (e.g. camelCase, PascalCase, under_score), which were studied in [7-9] using eye movement parameters in the case of source codes regarding clarity and readability. In addition to all these results, studies [10-13] found that reading is less linear and the duration of fixations is longer in case of the source codes than in traditional texts, however, study [14] also showed that the linearity of readability of these source codes can be increased and fixations duration can be reduced with the clean code technique. The result related to the coding technique is pointed out in the study [15], according to which beginner developers study source code comments longer to understand how the application works than their experienced counterparts. The [16] article used different visual elements to increase the efficiency of clone code detect.

The application and effects of visual elements were also analysed in [17-19] research in the design phase with eye movement parameters using different layout and coloring techniques regarding UML class diagrams and their stereotypes [20] [21], Business Process Models (BPM) [22] and Entity Relationship Diagrams (ERD) [23].

Another difficult task of software development is to detect and fix bugs. In studies [15] [27] [28] different approaches to examine and increase the efficiency of these processes using eye movement tracking systems were used. In research [25] [26] two different methods of error detection and correction were compared. In addition to the efficiency analysis, there was a significant difference in the number of fixations in case of those subjects who focused only on a small part of the source code compared to those who tried to understand the whole code and made some important bug corrections [25, 26].

Research [31] provides an overview of the interdisciplinary combination of educational science, psychology, software engineering and the eye tracking methodology. It also defines that the domain of software engineering offers great potential for applied eye tracking research and in turn, it can benefit from the possibilities of this upcoming technology as well.

Paper [32] compares only complete query expressions in different syntax ways. Based on the results of the applied knowledge level assessment tests, the test subjects determined the results of each query faster and more accurately on average. According to the duration and amount of fixation as well as the saccade duration and length, in case of the method syntax, information retrieval is more difficult, the search is less efficient, there is more uncertainty, and a greater mental effort is required to determine the results of the queries. Having finished the research of [32], in this paper, it was repeated with other test subjects and it was found examining each operation in more detail that the results are not as clear as in case of previous study.

The aim of this paper is to examine the readability and comprehensibility of the two options of the Language-Integrated-Query (LINQ) abstraction layer that appeared in C# 3.0, the query syntax and the method syntax including each operation, by statistically evaluating eye movement parameters in addition to traditional knowledge level tests and questionnaires taking into account Microsoft statement [29] [30], according to which semantically identical but syntactically different queries are easier to read and are interpreted more effectively than the method syntax-based queries.

In Section 2, the hypotheses, the applied devices and source codes, the process and the environment of the examination as well as the data of the test subjects are summarized. In Section 3, the results obtained during the research are analysed and evaluated. In Section 4, conclusions based on the obtained results are drawn, while in Section 5, the study is summarized and conclusions are drawn.

2 Research Goal and Applied Methodology

2.1 Materials

During the tests, the eye movement parameters of the test subjects were observed, recorded and evaluated, with a general-purpose research grade device, the GP3 eye-tracker, which is a portable device that 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, with the applications required to operate the device, as well as with an open source coded, free software package, the OGAMA

(OpenGazeAndMouseAnalyter). The results were also stored in a database for further statistical use. The applied devices and software have already been used successfully in other scientific research [27] [28] [33-36]. During the study, the test subjects received the same queries in random order, a total of 10-10 pieces from both syntax types, and the possible results had to be determined regarding them. The examination algorithms were readable in the Visual Studio development environment, which filled the computer screen as much as possible. To display each query, an LG 22M45 1920x1080 resolution, 22" diameter monitor, with the maximum resolution set was used. Figure 1 shows a schematic diagram of the equipment setup.

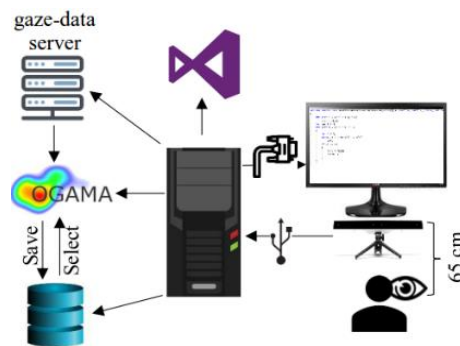


Figure 1

A schematic diagram of the equipment setup [32]

2.2 Procedure

During eye movement recording, the human eyes often show unwanted noise or baseline drift due to fatigue or distraction, which interfere with the analysis of signals, therefore, the devices and the test subjects were not exposed to direct, disturbing sunlight as it would interfere with the reading of corneal reflexes, and the audio-visual stimuli was minimized, so the subjects were helped to reach the highest possible concentration by excluding possible disturbing factors. The best calibration results were accepted for each test subject before each procedure. In case a test subject wore glasses, it was best to tilt the GP3 unit upwards at a greater angle to prevent reflections from appearing off the glasses lens. As only static images had to be observed during the examination, any interference that might have occurred was negligible.

A syntactically different but semantically equivalent query could not be observed by the same test subject due to the acquisition of the source code output, therefore, a different but equally difficult series of tests were used. In Figure A1, an example of the presented stimulus is shown.

Furthermore, the queries appeared on the screen in a random order, each of which could be studied for a maximum of 60 seconds, and then their possible outcome had to be determined. During the study of the queries, the eye movement parameters were continuously monitored and recorded. After determining the possible result of the last query, the recorded gaze route metrics were saved in a database for further use and statistical evaluation. After the successful data backup, during an interview, the test subjects had to complete a questionnaire about the test. The process describing the test is briefly summarized in Figure 2.

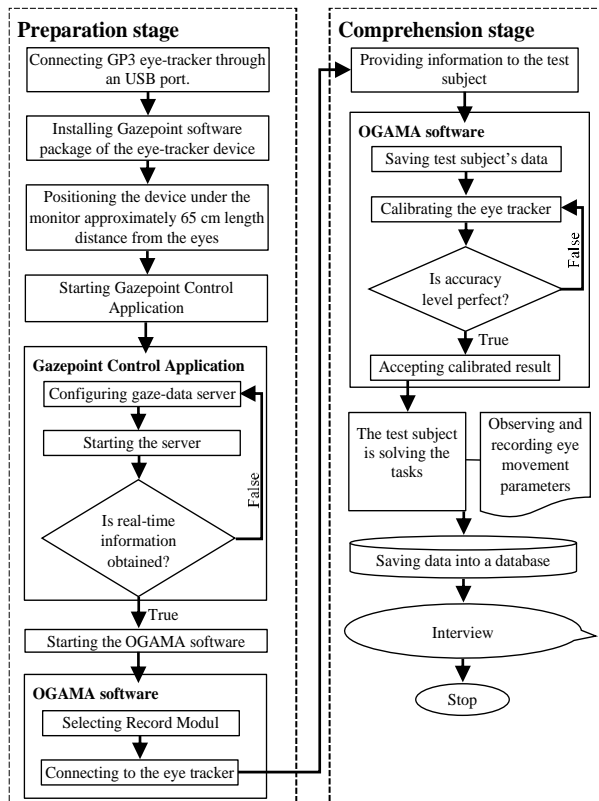


Figure 2

A schematic diagram of the equipment setup [32]

2.3 Participant

17 females and 40 males, a total of 57 university students, between the ages of 19 and 22 ($M=20.5$, $SD=1.19$), volunteered for the research, signing a consent form, who declared themselves to be completely healthy and did not know of any disease, nor, were they under medical influence or treatment. Another condition of the application was that the courses related to programming that provided the

knowledge necessary for the performance of the task had to be completed. In order to avoid the impact on the time-based statistics results, a preliminary test was performed, where the reading speed of the test subjects was examined. They read an informational text about the expected tasks and as a result, significant difference in reading time could not be detected ($p>0.05$).

2.4 Applied Algorithms

During the research, the most varied LINQ queries possible for data from 3 different data sources were compiled (LINQ To Objects, LINQ To XML and LINQ To Entities). Each query syntax-based query included the from and select clauses, which additional clauses were supplemented (*where*, *let* or *orderby*) in such a way that one clause more emphatically was included than the others. This method was also used for method syntax-based queries. In order to minimize the distortion of the eye movement parameters articulating the source code in case of both query types, all query operators and method calls as well as their parameterizations were listed in a row.

2.5 Hypotheses Formulation

To execute the research, a data abstraction layer was used that provides a unified language for the structured and type-safe processing of data from different data sources (objects - LINQ To Objects, relational databases - LINQ To SQL, XML - LINQ To XML, etc.). The query forms of LINQ technology, the query syntax and method syntax, released in C#3.0, also make this type of data processing possible. Despite the claims according to which there is no semantic or performance difference between the two types of LINQ queries, the Common Language Runtime (CLR) translates query syntax-based queries into method calls. When the code is compiled, some operations are only available as method calls. The method syntax also fits better into the source code world specific of C#, with query expressions being more common because they are better readable and easier to understand than their equivalent method syntax-based ones. [29] [30]

The aim of this research is to examine the readability and comprehensibility of the two query types, involving test subjects, by statistically evaluating eye movement parameters in addition to traditional knowledge level test, taking into account Microsoft's claim [29] [30] that semantically the same but different in syntax query syntax-based queries are easier for people to read and understand more effectively than method syntax-based ones. Previous research has determined that a longer fixation period may indicate difficulty in retrieving information, a larger fixation amount may indicate less efficient search and greater uncertainty, smaller saccades in the period may indicate greater complexity of the tasks, and shorter saccades may indicate greater mental strain. [37-42].

Based on the above, five hypotheses were formulated:

H₁: Based on the statistical evaluation of the knowledge level assessment results, the test subjects were able to determine the results of the query syntax-based queries more accurately and faster than the method syntax-based queries and are these differences significant taking into account each query operators and their equivalent extended method calls with semantically identical parameterizations in details?

H₂: Fixation durations are longer on average, so information retrieval is more difficult for method syntax-based LINQ queries than for query syntax-based queries and is this difference significant taking into account each query operators and their equivalent extended method calls with semantically identical parameterizations in details?

H₃: The amount of fixation is higher on average, so is the search less efficient and is the uncertainty higher for method syntax-based LINQ queries than for query syntax-based ones and is this difference significant taking into account each query operators and their equivalent extended method calls with semantically identical parameterizations in details?

H₄: The average saccade duration is less, so the tasks seem more complex in case of method syntax-based LINQ queries than in case of query syntax-based ones and is this difference significant taking into account each query operators and their equivalent extended method calls with semantically identical parameterizations in details?

H₅: The average saccade distance is smaller, so the mental load is higher in case of method syntax-based LINQ queries than in case of query syntax-based ones and is this difference significant taking into account each query operators and their equivalent extended method calls with semantically identical parameterizations in details?

3 Results

The first step of the evaluation of the results was the knowledge test evaluation, in which case it was defined how quickly and accurately the test subjects could determine the results of each query type. At the second stage, different eye movement parameters were examined and rated. In selecting the appropriate statistical tests, it was considered that the test subjects were independent of each other and examined the same subjects within a group. It was also determined that for the examination of the distribution of the variables, the Shapiro-Wilk test would be applied.

3.1 The Evaluation of Knowledge Level Test Results

During the evaluation of the test-based results, it was found that all test subjects formulated a possible output and did not run out of the maximum of 60 seconds per question time limit. In case of both query and method syntax, the worst result available was 10, while the best result included 0 incorrect answers. The evaluation of the test results is summarized in Table 1.

Table 1

The maximum, minimum and average number of incorrect answers and their dispersion in case of the query and method syntax ($n=57$)

Query Syntax				Method Syntax			
Min	Max	Mean	SD	Min	Max	Mean	SD
0	4	1.86	1.27	0	6	2.82	1.30

The normality results of incorrect answers are also significant in case of both query syntax ($W(57)=0.880$, $p<0.001$) and method syntax ($W(57)=0.893$, $p<0.001$), therefore, using paired-samples Wilcoxon-test ($T=319$, $Z=-3.269$, $p=0.001$ (2-tailed), $r=0.433$), it can be stated that in the case of the method syntax ($Mdn=2$) with moderate effect, but significantly more erroneous outputs were formulated than in case of the query syntax ($Mdn=2$). The distribution of the incorrect answers is shown in Figure 3a.

The worst time available was also 60 seconds per question, a total of 600 seconds in case of both the query and method syntax. The evaluation of the time required to perform the tests is summarized in Table 2.

Table 2

The maximum, minimum and average number of incorrect answers and their dispersion in case of the query and method syntax ($N=57$)

Query Syntax				Method Syntax			
Min	Max	Mean	SD	Min	Max	Mean	SD
288	386	335.86	22.35	344	422	386.49	18.54

The normality results of the durations query syntax ($W(57)=0.977$, $p=0.347$) and method syntax ($W(57)=0.981$, $p=0.498$) are not significant, therefore, using a paired-samples t-test ($t(56)=-12.74$, $p<0.001$ (2-tailed), $r=0.862$), it can be stated that the results of the query syntax ($M=335.86$, $SD=22.35$) queries were determined in a significantly shorter time with a marked effect than in case of the method syntax ($M=386.49$, $SD=18.54$). The confidence interval of the average time to complete, is shown in Figure 3b.

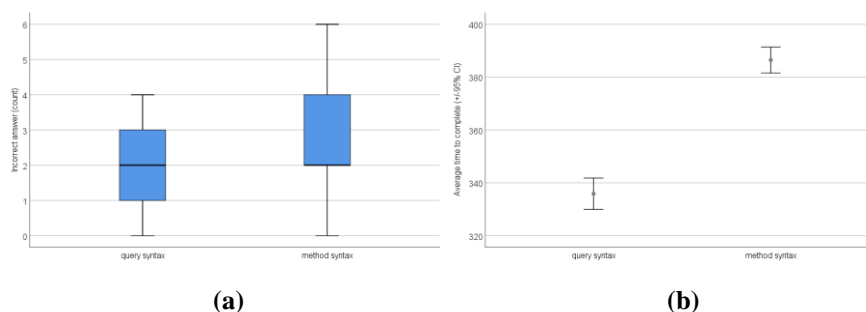


Figure 3

The distribution of incorrect answers (a) and the confidence interval of average time to complete (b)

During the further evaluation of the knowledge level assessment results, it was also examined query operators and their equivalent extended method calls, as in case of each query, one operator or method call was more prominent. Table 3 shows the average of the error numbers and the average time required to determine the output of the query for the entire sample.

Table 3

The average number of erroneous responses broken down into query operators and method calls for query and method syntax ($N=57$)

Q ¹ Query Operator ²	Query Syntax				Extension method ²	Method Syntax			
	Number of errors		Time to complete (s)			Number of errors		Time to complete (s)	
	Mean	SD	Mean	SD		Mean	SD	Mean	SD
1 join	0.28	0.45	38.42	9.70	Join	0.42	0.50	50.75	7.46
2 join	0.26	0.44	41.70	8.79	Join	0.46	0.50	54.18	4.97
3 let	0.21	0.41	40.49	9.36	Select	0.37	0.49	44.54	4.39
4 let	0.18	0.38	39.72	8.61	Select	0.39	0.49	46.58	4.93
5 join	0.25	0.43	39.88	5.89	Join	0.40	0.49	49.98	6.13
6 group by	0.14	0.35	28.35	4.38	GroupBy	0.18	0.38	27.58	5.89
7 orderby with multiple orders	0.09	0.29	23.05	4.23	OrderBy ThenBy	0.11	0.31	27.49	4.05
8 where	0.05	0.23	19.05	5.50	Where	0.02	0.13	17.14	5.39
9 where	0.07	0.26	21.82	5.29	Where	0.05	0.23	19.86	5.47
10 let	0.33	0.48	43.37	8.45	Select	0.44	0.50	48.39	5.02

¹The serial number of the question.

²Prominent query operators and their equivalent extended method calls with semantically identical parameterization.

Answering the H_1 hypothesis, during the statistical evaluation of the results it can be determined that in the case of the knowledge level test the subjects were able to determine the results of the query syntax-based queries more accurately and faster than those of the method syntax-based queries and these differences are significant ($p < 0.05$) taking into account that when they had to determine the possible outputs for shorter method syntax-based queries containing the *Where* method call, more

accurate ($M=0.035$, $SD=0.18$) incorrect answers and faster ($M=18.5$, $SD=5.43$ seconds) results were obtained than in case of where query operator ($M=0.06$, $SD=0.245$) incorrect responses and ($M=20.44$, $SD=5.4$ seconds). In contrast, query syntax proved to be more efficient for more complex as well as *let* ($M=0.24$, $SD=0.42$) pieces of erroneous responses and ($M=41.19$, $SD=8.81$ seconds), and *join* ($M=0.26$, $SD=0.44$) pieces of erroneous responses and ($M=40$, $SD=8.13$ seconds) in case of queries containing query operators such as *Select* ($M=0.4$, $SD=0.493$) incorrect answers and ($M=46.5$, $SD=4.78$ seconds), and *Join* ($M=0.43$, $SD=0.5$) pieces of erroneous responses and ($M=51.64$, $SD=6.19$ seconds) during method calls.

3.2 Evaluation based on Eye Movement Parameters

Because the aim of the research was to compare the readability and intelligibility of the two query types, the Area of Interest (AOI) areas were selected for each source code that included only the queries and writing before evaluating eye movement parameters with proper source code segmentation. Since 10 queries were formulated for both the query and method syntaxes, the same number of AOIs, a total of 10-10 were selected. Within the AOI areas, the total fixation duration and amount, the average saccade duration and distance were recorded and evaluated in case of each test subject.

In the further evaluation of the eye movement parameters, the query operators and their equivalent extended method calls were also examined, as in case of each query one operator or method call was emphasized, and during answering the H_1 hypothesis, it was found that the readability and intelligibility of the query or method syntax depend on the complexity of the query and the applied query operators, as well as its equivalent method calls. Because each AOI areas contain a query, a query operator or extended method is given more emphasis in each AOI area.

3.2.1 Complete Fixation Time based on AOIs

The normality results of the averages of the total fixation time measured in all AOI areas are not significant query syntax ($W(57)=0.965$, $p=0.099$) and method syntax ($W(57)=0.978$, $p=0.366$), so a paired sample t-test was used ($t(56)=-16,192$, $p<0.001$ (2-tailed), $r=0.908$), based on which it can be stated that the mean total fixation time was significantly longer with a marked effect in case of the method syntax ($M=23296.867$, $SD=1447.381$) milliseconds than in case of the query syntax ($M=19128.118$, $SD=1553.322$) milliseconds. The confidence interval of complete fixation time based on AOIs is shown in Figure 4.

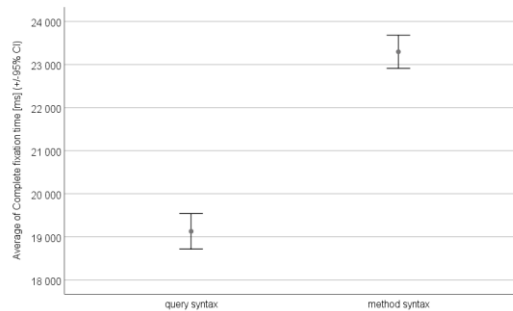


Figure 4

The confidence interval average of complete fixation time based on AOIs

Table 4 shows the statistical evaluation of the query operators and extended methods with semantically identical parameterization in each AOI area.

Table 4

The statistical evaluation of query operators and extended methods with semantically identical parameterization in each AOI area in case of complete fixation time ($N=57$)

query operator ² extension method ²	Median (milliseconds)	T	Z	<i>p</i> value	<i>r</i>
where Where	9635.50 8452.50	1979.50	-3.670	<0.001	0.34 ¹
join Join	21786.00 30381.00	1763.50	-8.621	<0.001	0.66 ¹
let Select	26472.00 31864.00	2710.00	-7.161	<0.001	0.54 ¹
groupby GroupBy	11413.00 11193.00	762.00	-0.512	0.608	0.07 ¹
orderby with multiple orders OrderBy and ThenBy	15232.00 15949.00	632.50	-1.541	0.123	0.20 ¹

¹large practical significant difference ($r \geq 0.5$).

²Prominent query operators and their equivalent extended method calls with semantically identical parameterization.

It can be concluded that with a marked effect ($r \geq 0.5$) the average fixation time was significantly ($p < 0.001$) longer in the *Join* ($Mdn=30381$), *Select* ($Mdn=31864$) extension methods than in case of the *join* ($Mdn=21786$), *let* ($Mdn=26472$) query operators, and with medium effect ($r \geq 0.3$) in case of the *where* ($Mdn=9635.5$) query operator and *Where* ($Mdn=8452.5$) extension method, and the grouping *groupby* query operator ($Mdn=11413$) and *GroupBy* extension method ($Mdn=11193$) and complex sorting operations *orderby* query operator ($Mdn=15232$) and *OrderBy* and *ThenBy* extension methods ($Mdn=15949$) no significant difference was detected ($p > 0.05$).

In response to the H_2 hypothesis, it can be stated that based on the obtained results, the average fixation time is longer in case of the method syntax-based

LINQ queries, however, information retrieval proved to be more difficult than in case of the syntax-based queries, and this difference is significant ($p < 0.001$), considering that in the case of shorter method syntax-based queries, containing only filtering condition and the lambda expression, the fixation time with medium effect ($r = 0.34$) is significantly ($p < 0.001$), shorter than in the case of the query syntax-based operations, furthermore, in the case of the grouping and complex sorting operations significant difference cannot be detected ($p > 0.05$).

3.2.2 Number of Fixations based on AOIs

The normality results of the average fixation numbers measured in all AOI areas are not significant in case of the query syntax ($W(57) = 0.980$, $p = 0.454$) and method syntax ($W(57) = 0.983$, $p = 0.621$), therefore, a paired sample t-test was used ($t(56) = -15.051$, $p < 0.001$ (2-tailed), $r = 0.895$), based on which it can be stated that with a marked effect the fixation amount is significantly less in case of the query syntax ($M = 42.296$, $SD = 2.905$) numbers than in case of the method syntax ($M = 52.988$, $SD = 4.206$) numbers. The confidence interval average of the number of fixations based on AOIs is shown in Figure 5.

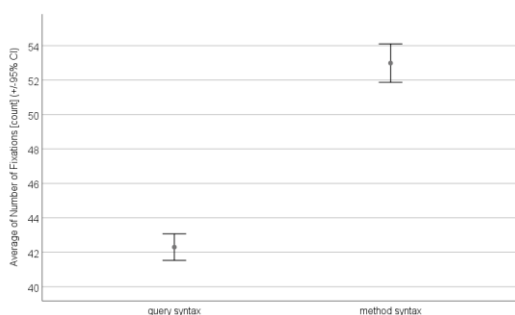


Figure 5

The confidence interval average of number of fixations based on AOIs

Table 5 shows the statistical evaluation of the query operators and extended methods with semantically identical parameterization in each AOI area.

Table 5

The statistical evaluation of query operators and extended methods with semantically identical parameterization in each AOI area in case of the number of fixation ($N = 57$)

query operator ² extension method ²	Median (numbers)	T	Z	p value	r
where Where	29.00 21.00	748.00	-7.087	<0.001	0.47 ¹
join Join	50.00 70.00	924.50	-9.825	<0.001	0.53 ¹
let Select	50.00 65.00	1315.00	-9.160	<0.001	0.50 ¹

groupby	29.00				
GroupBy	32.00	614.00	-0.460	0.646	0.04 ¹
orderby with multiple orders	39.00				
OrderBy and ThenBy	41.00	602.00	-1.409	0.159	0.13 ¹

¹large practical significant difference ($r \geq 0.5$).

²Prominent query operators and their equivalent extended method calls with semantically identical parameterization.

It can be concluded that with a marked effect ($r \geq 0.5$) the average fixation amount was significantly ($p < 0.001$) higher in the *Join* ($Mdn=70$) and *Select* ($Mdn=65$) extension methods than in case of the *join* ($Mdn=50$) and *let* ($Mdn=50$) query operators. With medium effect ($r \geq 0.3$) in case of the *Where* ($Mdn=21$) extension method and *where* ($Mdn=29$) query operator as well as the *groupby* query operator ($Mdn=29$) and *GroupBy* extension method ($Mdn=32$) and in case of the complex sorting operations *orderby* query operator ($Mdn=39$), *OrderBy* and *ThenBy* extension methods ($Mdn=41$) no significant difference was detected ($p > 0.05$).

In response to the H_3 hypothesis, based on the obtained results, it can be claimed that the average fixation amount is more in case of the method syntax-based LINQ queries, search is less efficient and uncertainty is higher than in case of the query syntax-based ones, and this difference is significant ($p < 0.001$) considering that in the case of shorter-term method syntax-based queries containing only the filter condition and the lambda expression, the fixation amount with medium effect ($r=0.47$) is significantly ($p < 0.001$) less than in case of the query syntax-based queries, and in case of the operations executing complex sorting no significant difference can be detected ($p > 0.05$).

3.2.3 Saccade Durations Mean based on AOIs

As the results of the average saccade duration normality measured in all AOI areas in query syntax ($W(57)=0.816$, $p < 0.001$) and method syntax ($W(57)=0.941$, $p=0.007$) are significant, so a paired-samples Wilcoxon-test was used ($T=32$, $Z=-6.312$, $p < 0.001$ (2-tailed), $r=0.591$), based on which it can be defined that the mean saccade duration mean was significantly shorter with a marked effect in case of the method syntax ($Mdn=185.57$) milliseconds than in case of the query syntax ($Mdn=210.09$) milliseconds. The confidence interval of the average of saccade duration mean based on AOIs is shown in Figure 6.

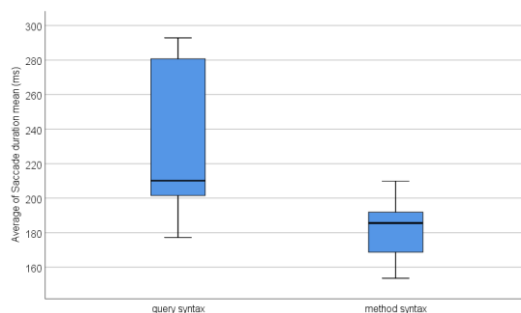


Figure 6

The distribution of the average of saccade duration mean based on AOIs

Table 6 shows the statistical evaluation of the query operators and extended methods with semantically identical parameterization in each AOI area.

Table 6

The statistical evaluation of query operators and extended methods with semantically identical parameterization in each aoi area in case of the saccade duration mean ($n=57$)

query operator ² extension method ²	Median (milliseconds)	T	Z	<i>p</i> value	<i>r</i>
where Where	258.50 275.00	2384.00	-2.526	0.012	0.16 ¹
join Join	218.00 135.00	125.50	-11.113	<0.001	0.60 ¹
let Select	206.00 118.00	1.00	-11.340	<0.001	0.61 ¹
groupby GroupBy	220.00 213.00	738.50	-0.485	0.627	0.05 ¹
orderby with multiple orders OrderBy and ThenBy	267.00 282.00	1056.00	-1.824	0.068	0.17 ¹

¹large practical significant difference ($r \geq 0.5$).

²Prominent query operators and their equivalent extended method calls with semantically identical parameterization.

It can be concluded that with marked effect ($r \geq 0.5$) the mean saccade duration was significantly ($p < 0.001$) longer in the *Join* ($Mdn=135$), *Select* ($Mdn=118$) extension methods than in *join* ($Mdn=218$), *let* ($Mdn=206$) query operators. With small effect ($r \leq 0.3$) in case of the *Where* ($Mdn=275$) extension methods and *where* query operators ($Mdn=258.5$) significant difference was detected ($p=0.012$). In case of grouping operations *groupby* query operators ($Mdn=220$) and *GroupBy* extension methods ($Mdn=213$) and complex sorting operations *orderby* ($Mdn=267$) and *OrderBy* and *ThenBy* extension methods ($Mdn=282$) the mean saccade duration was not significantly longer ($p > 0.05$).

In response to the H_4 hypothesis, it can be stated that based on the results obtained, the average saccade duration is less and the tasks are more complex and more difficult to solve in method syntax-based LINQ queries than in case of the

query syntax-based ones, and this difference is significant ($p < 0.001$) considering that in the case of method syntax-based queries containing only a filter condition and a lambda expression, the saccade duration with small effect ($r \leq 0.3$) is significantly ($p < 0.001$) more than in case of query syntax-based queries, moreover, significant difference cannot be detected in grouping operations ($p > 0.05$).

3.2.4 Saccade Distance Mean based on AOIs

The normality results of the average saccade length measured in all AOI areas are not significant query syntax ($W(57)=0.975$, $p=0.299$) and method syntax ($W(57)=0.985$, $p=0.685$), therefore, a paired sample t-test was used ($t(56)=5.688$, $p < 0.001$ (2-tailed), $r=0.605$), based on which it can be stated that significantly shorter saccades can be measured in case of the method syntax ($M=194.16$, $SD=11.44$) pixels than in case of the query syntax ($M=207.27$, $SD=12.69$) pixels. The confidence interval of average of saccade distance mean based on AOIs is shown in Figure 7.

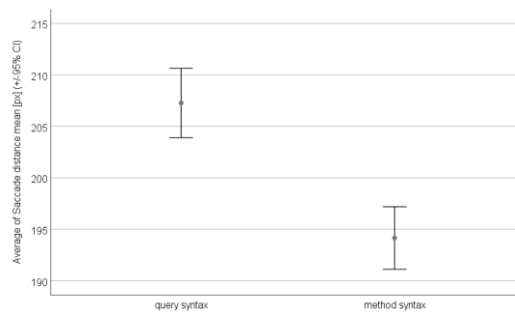


Figure 7

The distribution of the average of saccade duration mean based on AOIs

Table 7 shows the statistical evaluation of the query operators and extended methods with semantically identical parameterization in each AOI area.

Table 7

The statistical evaluation of query operators and extended methods with semantically identical parameterization in each AOI area concerning saccade distance mean ($N=57$)

query operator ² extension method ²	Median (pixels)	T	Z	p value	r
where	140.50	2253.50	-2.771	0.006	0.26 ¹
Where	154.00				
join	268.00	2315.00	-7.771	<0.001	0.59 ¹
Join	218.00				
let	241.00	8726.50	-2.119	0.034	0.16 ¹
Select	223.00				
groupby	176.00	639.00	-1.490	0.136	0.20 ¹
GroupBy	155.00				

orderby with multiple orders	133.00	656.50	-1.351	0.177	0.18 ¹
OrderBy and ThenBy	128.00				

¹large practical significant difference ($r \geq 0.5$).

²Prominent query operators and their equivalent extended method calls with semantically identical parameterization.

It can be concluded that with marked effect ($r \geq 0.5$) the average saccade length was significantly ($p < 0.05$) shorter in case of *Join* ($Mdn=218$) extension method than in case of *join* ($Mdn=268$) query operator, and with small effect ($r \leq 0.3$) in case of the *Select* ($Mdn=223$) extension method and *where* ($Mdn=140.5$) query operator it was shorter than in case of *let* ($Mdn=241$) query operator and *Where* ($Mdn=154$) extension method, furthermore, in case of grouping operators *groupby* query operator ($Mdn=176$) and *GroupBy* extension method ($Mdn=155$) and complex sorting operations (*orderby* query operator ($Mdn=133$) and *OrderBy* and *ThenBy* extension methods ($Mdn=128$)) no significant difference was detected ($p > 0.05$).

In response to the H_5 hypothesis, it can be stated that based on the obtained results, the average saccade length in case of method syntax-based LINQ queries is shorter and it can be read and interpreted with higher mental load than in case of the query syntax-based ones and this difference is significant ($p < 0.05$) considering that in the case of shorter method syntax-based queries containing only a filter condition and a lambda expression, the average saccade distance with medium effect ($r=0.26$) is significantly ($p=0.006$) longer than that of the query syntax-based queries, moreover, in case of the grouping and complex sorting operators no significant difference was detected ($p > 0.05$).

4 Discussion

The results of the research show that the evaluated knowledge level test and eye movement parameters, are in accordance, but conversely, the results of the university students support Microsoft's claim [29] [30], that the semantically identical, but different in syntax query syntax-based queries, are easier to read and interpret more effectively, than method syntax-based queries. Based on the results of the knowledge tests, the test subjects determined the results of each query faster and more accurately, on average. According to the duration and amount of fixation and the duration and length of the saccade, in case of the method syntax, information retrieval is more difficult, the search is less efficient, there is more uncertainty, and overall, a greater mental effort is required to determine the results of the queries.

However, if the results are examined in more detail, the statements above cannot be defined so clearly, as query syntax-based queries are more readable and easier to interpret than method syntax-based queries using multiple *join* and *let*

operations. In contrast, using shorter specifically *where* operations and lambda expressions, method syntax-based queries are easier to decipher with lower mental load. As a result of the study, it can also be stated, that the intelligibility of the grouping and sorting operations, the two syntactic types, is not significantly different from each other.

As a further result of the research, it can be stated that source code analysis with eye movement tracking, can provide more accurate and precise results, than the traditional knowledge level test, because by marking AOI areas, source code details, can be more clearly examined. As a result, we are not only able to evaluate the readability and comprehensibility of a code base as a whole, but also, in a targeted way, so more difficult parts can be explored in detail. Furthermore, the importance of code readability and interpretability can be paramount not only when comparing individual technologies but in case of the whole code, furthermore, it can provide additional information during debugging processes. The digital culture subject [43] can be considered as an effective foundation subject, the application of which can contribute to increasing the efficiency of IT learning.

All in all, the use of eye movement tracking systems can be a possibility to analyse the effectiveness of education [44-46], learning and different programming technologies. The eye-tracking systems allows to measure the direction of the gaze and to record the movement of the eye relative to the head it means that systems are well suited as well for studying the cognitive processes of individuals with mental disorders [47]. These systems are being used successfully in more and more fields of science. The paper has an important contribution to CogInfoCom [48] [49] research field as well. Moreover, exploring the gaze route in virtual environments [50] [51] related to digital information [52-54] or education [55] [56] can contribute to the development of more effective curricula.

Conclusions

The aim of this paper was to analyse the readability and comprehensibility of the query syntax and method syntax, including each operation, the two possibilities of the Language-Integrated-Query (LINQ) data abstraction layer that appeared in C# 3.0, by using traditional knowledge level tests and eye movement metrics.

Based on the results of the applied knowledge level test, it can be stated that, the test subjects were able to determine the correct results of the query syntax-based queries significantly more accurately and faster than those of the method syntax-based queries. The test results, assessing the level of knowledge, were also confirmed by the eye movement parameters, as the total fixation duration is significantly longer, the average number of fixations is significantly higher, the average saccade duration is significantly less and the average saccade length is significantly shorter, in case of the method syntax-based queries and the query syntax-based queries.

However, examining each query in more detail, the results are not so clear, as there are cases, where the method syntax is more efficient, as well as, operations where no significant difference can be detected between the two types of queries. The results of the research show that the source code analysis with eye movement tracking can provide more accurate and precise results than the traditional knowledge level test, because by marking AOI areas, source code details can be clearly examined. In case of the obtained results, it should be considered that the test subjects were university students who did not have more serious programming experience. In the future, further investigation of the two query types, will be required, for different query characteristics and properties, as well as, experienced test subjects.

References

- [1] T. Ujbanyi, A. Kovari, G. Sziladi, J. Katona: Examination of the Eye-Hand Coordination Related to Computer Mouse Movement, *Infocommunications Journal*, Vol. 12, 2020, pp. 26-31
- [2] K. Bortko, P. Bartków, J. Jankowski, D. Kuras, P. Sulikowski: Multi-Criteria Evaluation of Recommending Interfaces towards Habituation Reduction and Limited Negative Impact on User Experience. *Procedia Computer Science*, Vol. 159, 2019, pp. 2240-2248
- [3] P. Sulikowski, T. Zdziebko: Deep Learning-Enhanced Framework for Performance Evaluation of a Recommending Interface with Varied Recommendation Position and Intensity Based on Eye-Tracking Equipment Data Processing, *Electronics*, Vol. 9, No. 2, 2020, pp. 1-15
- [4] P. Sulikowski: Evaluation of Varying Visual Intensity and Position of a Recommendation in a Recommending Interface Towards Reducing Habituation and Improving Sales. In *Proceedings of the International Conference on e-Business Engineering*; Springer, 2019; pp. 208-218
- [5] J. Epelboim, J. R. Booth, R. Ashkenazy, A. Taleghani, R. M. Steinman: Fillers and spaces in text: The importance of word recognition during reading. *Vision research*, Vol. 37, 1997, pp. 2899-2914
- [6] K. Rayner, M. H. Fischer, A. Pollatsek: Unspaced text interferes with both word identification and eye movement control. *Vision Research*, Vol. 38, 1998, pp. 1129-1144
- [7] B. Sharif, J. I. Maletic: An eye tracking study on camelcase and under_score identifier styles. In *Proceedings of the 2010 IEEE 18th International Conference on Program Comprehension*; IEEE, 2010; pp. 196-205
- [8] F. Deissenboeck, M. Pizka: Concise and consistent naming. *Software Quality Journal*, Vol. 14, 2006, pp. 261-282

-
- [9] D. Binkley, M. Davis, D. Lawrie, C. Morrell: To camelcase or under_score. In Proceedings of the 2009 IEEE 17th International Conference on Program Comprehension; IEEE, 2009; pp. 158-167
- [10] M. E. Crosby, J. Stelovsky: How do we read algorithms? A case study. *Computer*, Vol. 23, 1990, pp. 25-35
- [11] M. E. Crosby, J. Scholtz, S. Wiedenbeck: The Roles Beacons Play in Comprehension for Novice and Expert Programmers. In Proceedings of the PPIG; Citeseer, 2002; p. 5.
- [12] T. Busjahn, C. Schulte, A. Busjahn: Analysis of code reading to gain more insight in program comprehension. In Proceedings of the Proceedings of the 11th Koli Calling International Conference on Computing Education Research, 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. In Proceedings of the Proceedings of the Symposium on Eye Tracking Research and Applications, 2014; pp. 335-338
- [14] J. Katona: Clean and Dirty Code Comprehension by Eye-tracking Based Evaluation using GP3 Eye Tracker. *Acta Polytechnica Hungarica*, Vol. 18, No. 1, 2021, pp. 79-99
- [15] S. Seha, G. Papangelakis, D. Hatzinakos, A. S. Zandi, F. J. Comeau: Improving Eye Movement Biometrics Using Remote Registration of Eye Blinking Patterns. In Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); IEEE, 2019; pp. 2562-2566
- [16] M. S. Uddin, V. Gaur, C. Gutwin, C. K. Roy: On the comprehension of code clone visualizations: A controlled study using eye tracking. In Proceedings of the 2015 IEEE 15th International Working Conference on Source Code Analysis and Manipulation (SCAM); IEEE, 2015; pp. 161-170
- [17] B. De Smet, L. Lempereur, Z. Sharafi, Y.-G. Guéhéneuc, G. Antoniol, N. Taube Habra: Visualizing and analyzing eye-tracking data. *Science of Computer Programming*, Vol. 79, 2014, pp. 260-278
- [18] G. C. Porras, Y.-G. Guéhéneuc: An empirical study on the efficiency of different design pattern representations in UML class diagrams. *Empirical Software Engineering*, Vol. 15, 2010, pp. 493-522
- [19] S. Jeanmart, Y.-G. Gueheneuc, H. Sahraoui, N. Habra: Impact of the visitor pattern on program comprehension and maintenance. In Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement; IEEE, 2009; pp. 69-78

- [20] S. Yusuf, H. Kagdi, J. I. Maletic: Assessing the comprehension of UML class diagrams via eye tracking. In Proceedings of the 15th IEEE International Conference on Program Comprehension (ICPC'07); IEEE, 2007; pp. 113-122
- [21] B. Sharif, J. I. Maletic: An eye tracking study on the effects of layout in understanding the role of design patterns. In Proceedings of the 2010 IEEE International Conference on Software Maintenance; IEEE, 2010; pp. 1-10
- [22] R. Petrusel, J. Mendling: Eye-tracking the factors of process model comprehension tasks. In Proceedings of the International Conference on Advanced Information Systems Engineering; Springer, 2013; pp. 224-239
- [23] 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, 2013, pp. 2184-2195
- [24] B. Sharif, M. Falcone, J. I. Maletic: An eye-tracking study on the role of scan time in finding source code defects. In Proceedings of the Proceedings of the Symposium on Eye Tracking Research and Applications; 2012; pp. 381-384
- [25] J. Katona, A. Kovari, C. Costescu, A. Rosan, A. Hathazi, I. Heldal, C. Helgesen, S. Thill, R. Demeter: The Examination Task of Source-code Debugging Using GP3 Eye Tracker. In Proceedings of the 2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom); IEEE, 2019; pp. 329-334
- [26] A. Kovari, J. Katona, C. Costescu: Evaluation of Eye-Movement Metrics in a Software Debugging Task using GP3 Eye Tracker. *Acta Polytechnica Hungarica*, Vol. 17, No. 2, 2020, pp. 57-76
- [27] J. Meng, T. Streitz, N. Gulachek, D. Suma, B. He: Three-Dimensional Brain-Computer Interface Control Through Simultaneous Overt Spatial Attentional and Motor Imagery Tasks. *IEEE Transactions on Biomedical Engineering*, Vol. 65, 2018, pp. 2417-2427
- [28] J. Iskander, D. Jia, I. Hettiarachchi, M. Hossny, K. Saleh, S. Nahavandi, C. Best, S. Hosking, B. Rice, A. Bhatti, et al.: Age-Related Effects of Multi-screen Setup on Task Performance and Eye Movement Characteristics. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC); IEEE, 2018; pp. 3480-3485
- [29] Microsoft Query Syntax and Method Syntax in LINQ (C#). Available online: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/query-syntax-and-method-syntax-in-linq> (accessed on 26 April 2020)
- [30] Language-Integrated Query (LINQ) (C#). Available online: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/> (accessed on 26 April 2020)

-
- [31] F. Hauser, J. Mottok, H. Gruber: Eye Tracking Metrics in Software Engineering. In Proceedings of the Proceedings of the 3rd European Conference of Software Engineering Education; 2018; pp. 39-44
- [32] J. Katona, A. Kovari, I. Haldal, C. Costescu, A. Rosan, R. Demeter, S. Thill, T. Stefanut: Using Eye-Tracking to Examine Query Syntax and Method Syntax Comprehension in LINQ. In Proceedings of the 2020 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom); IEEE, 2020; pp. 437-444
- [33] B. J. Edelman, J. Meng, N. Gulachek, C. C. Cline, B. He: Exploring cognitive flexibility with a noninvasive BCI using simultaneous steady-state visual evoked potentials and sensorimotor rhythms. IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 26, 2018, pp. 936-947
- [34] A. Voßkühler, V. Nordmeier, L. Kuchinke, A. M. 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, 2008, pp. 1150-1162
- [35] T. Ujbanyi: Examination of eye-hand coordination using computer mouse and hand tracking cursor control. In Proceedings of the 2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom); IEEE, 2018; pp. 353-354
- [36] T. Ujbanyi, J. Katona, G. Sziladi, A. Kovari: Eye-tracking analysis of computer networks exam question besides different skilled groups. In Proceedings of the 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom); IEEE, 2016; pp. 277-282
- [37] M. A. Just: Carpenter, P.A. Eye fixations and cognitive processes. Cognitive psychology, Vol. 8, 1976, pp. 441-480
- [38] E. Stuyven, K. Van der Goten, A. Vandierendonck, K. Claeys, L. Crevits: The effect of cognitive load on saccadic eye movements. Acta psychologica, Vol. 104, 2000, pp. 69-85
- [39] S. Chen, J. Epps, N. Ruiz, F. Chen: Eye Activity as a Measure of Human Mental Effort in HCI. In Proceedings of the Proceedings of the 16th international conference on Intelligent user interfaces; 2011; pp. 315-318
- [40] J. Zagermann, U. Pfeil, H. Reiterer: Measuring Cognitive Load Using Eye Tracking Technology in Visual Computing. In Proceedings of the Proceedings of the sixth workshop on beyond time and errors on novel evaluation methods for visualization; 2016; pp. 78-85
- [41] M. Keskin, K. Ooms, A. O. Dogru, P. De Maeyer: Digital Sketch Maps and Eye Tracking Statistics as Instruments to Obtain Insights into Spatial Cognition. Journal of Eye Movement Research, Vol. 11, No. 3, 2018, pp. 1-20

- [42] M. Keskin, K. Ooms, A. O. Dogru, P. De Maeyer: EEG & Eye Tracking User Experiments for Spatial Memory Task on Maps. *ISPRS International Journal of Geo-Information*, Vol. 8, 2019, pp. 1-17
- [43] E. Karl and G. Molnar: Environmental education in light of the digital culture. *Journal of Applied Technical and Educational Sciences*, Vol. 10, No. 3, 2020, pp. 76-103
- [44] J. Francisti, et. al: Application Experiences Using IoT Devices in Education *Applied Sciences*, Vol. 10, No. 20, 2020, pp. 1-14
- [45] G. Molnar, et. al.: Solutions, experiences in online education in Hungary and Serbia related to the situation caused by Covid-19. In *Proceedings of the 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*; IEEE, 2020; pp. 603-607
- [46] B. Szenkovits, et. al.: Gamification and microcontent orientated methodological solutions based on bring-your-own device logic in higher education. In *Proceedings of the 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*; IEEE, 2018; pp. 385-388
- [47] C. Costescu and A. M. Rosan: Development an assessment protocol to identify the characteristics of ASD using eye-tracking for special education purpose. *Journal of Applied Technical and Educational Sciences*, Vol. 9, No. 4, 2019, pp. 70-87
- [48] P. Baranyi, A. Csapo, Gy. Sallai: *Cognitive Infocommunications (CogInfoCom)* Springer, 2015
- [49] P. Baranyi, A. Csapo: Definition and synergies of cognitive infocommunications. *Acta Polytechnica Hungarica*, 2012, Vol. 9, No. 1, pp. 67-83
- [50] T. Szakall, A. Sabo, B. Kuljic, and P. Elter: Implementing an existing teaching method in a virtual environment. *Journal of Applied Technical and Educational Sciences*. Vol. 9, No. 2, 2019, pp. 18-28
- [51] B. Berki: Desktop VR and the Use of Supplementary Visual Information. In *Proceedings of the 2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*; IEEE, 2018; pp. 333-336
- [52] B. Orosz, et. al.: Digital education in digital cooperative environments. *Journal of Applied Technical and Educational Sciences*, Vol. 9, No. 4, 2019, pp. 55-69
- [53] I. Horvath, A. Sudar: Factors contributing to the enhanced performance of the maxwhere 3d vr platform in the distribution of digital information. *Acta Polytechnica Hungarica*, 2018, Vol. 15, No. 3, pp. 149-173
- [54] I. Horváth: Evolution of teaching roles and tasks in VR / AR-based education. In *Proceedings of the 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*; IEEE, 2018; pp. 355-360

- [55] A. Kovari: CogInfoCom Supported Education: A Review of CogInfoCom Based Conference Papers. In Proceedings of the 2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom); IEEE, 2018; pp. 233-236
- [56] V. Kövecses-Gosi: Cooperative Learning in VR Environment. Acta Polytechnica Hungarica 2018, Vol. 15, No. 3, pp. 205-224

Appendix

In Figure A1, an example of the presented AOI stimulus is shown.

```
double minimumAverageScore = 70;
var qStudents =
    from student in students
    let averageScore = student.Scores.Average()
    let fullName = student.FirstName + " " + student.LastName
    where averageScore > minimumAverageScore
    orderby averageScore
    select new { StudentID = student.ID, FullName = fullName,
                AverageScore = averageScore };
```

(a)

```
double minimumAverageScore = 70;
var mStudents =
    students
    .Select(x => new { AverageScore = x.Scores.Average(), FullName
                    = x.FirstName + " " + x.LastName, x.ID })
    .Where(student => student.AverageScore > minimumAverageScore)
    .OrderBy(student => student.AverageScore)
    .Select(student => new { StudentID = student.ID,
                           student.FullName, student.AverageScore });
```

(b)

Figure A1

Two syntactically different (a: query syntax-based, b: method syntax-based) but semantically equivalent and equally difficult queries