

Article

# Measuring Cognition Load Using Eye-Tracking Parameters Based on Algorithm Description Tools

Jozsef Katona 

CogInfoCom Based LearnAbility Research Team, Department of Software Development and Application, Institute of Computer Engineering, University of Dunaujvaros, 2400 Dunaujvaros, Hungary; katonaj@uniduna.hu; Tel.: +36-25-551-605

**Abstract:** Writing a computer program is a complex cognitive task, especially for a new person in the field. In this research an eye-tracking system was developed and applied, which allows the observation of eye movement parameters during programming as a complex, cognitive process, and the conclusions can be drawn from the results. The aim of the paper is to examine whether the flowchart or Nassi–Shneiderman diagram is a more efficient algorithm describing tool for describing cognitive load by recording and evaluating eye movement parameters. The results show that the case of the interpreting flowchart has significantly longer fixation duration, more number of fixations, and larger pupil diameter than the case of the Nassi–Shneiderman diagram interpreting. Based on the results of the study, it is clear how important it is to choose the right programming tools for efficient and lower cost application development.

**Keywords:** cognitive task; cognition load; eye-tracking; programming; algorithm description tools



**Citation:** Katona, J. Measuring Cognition Load Using Eye-Tracking Parameters Based on Algorithm Description Tools. *Sensors* **2022**, *22*, 912. <https://doi.org/10.3390/s22030912>

Academic Editors: Enrico Vezzetti, Andrea Luigi Guerra, Gabriele Baronio, Domenico Speranza and Luca Ulrich

Received: 2 January 2022

Accepted: 22 January 2022

Published: 25 January 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Writing a computer program is a complex cognitive task, especially for a new person in the field [1]. In order to facilitate the process and to understand the algorithms to be implemented a higher level of algorithmic thinking and problem solving is required [2–4]. Algorithm description tools have been introduced to describe the algorithm to be implemented independently of the programming language in order to make the algorithms more transparent and understandable. The commonly used algorithm description tool is the flowchart (FCh), which describes the algorithm as a directed graph, thus illustrating the steps of execution, while the Nassi–Shneiderman diagram (NSD), also called NS diagram or structogram, represents the algorithm as a graph without edges as a “structured flow charts”. Additional algorithm description tools have recently been applied like the pseudocode, Jackson diagram, the sentence-like description and the description with tree; however, the current study performs the analysis of the previously mentioned FCh and NS diagrams.

In the visual programming described in the article by Charntaweekhun and Wangsiripitak [1], students learning programming can simply compile and run a program using the FCh without any coding and allow for easy debugging and detection. Thus, the presented system is excellent for teaching structural programming, as it avoids the learning difficulty caused by the individual syntax of programming languages, thus providing an opportunity to develop problem-solving skills [1,5].

In the study, Xinogalos [5] provides an overview of FCh-based programming environments and makes suggestions for making software engineering education more effective. In the article in Cabo [6], it was found that students who use FCh to solve problems effectively will learn Python or similar programming languages more easily ( $r$ -squared = 0.68). Therefore, the use of flowcharts in programming education could appear as a kind of supportive tool in the development of various cognitive processes [6]. Hooshyar et al. [7]

aimed to provide beginner programmers with a framework in which to develop their problem-solving skills, as they found that the weakness of this skill was directly related to the syntax of the programming language to be learned and the development environment used. However, the use of FCh can not only mean more efficient knowledge transfer in the field of education but can also be seen as a supportive tool in industry and in different areas of research.

Shafeek and Karunaratne [8] explain that in a modern distributed software development environment, communication between developers is mostly the source code, which is thus difficult in many cases. In the research, the authors implemented a software prototype that can turn source code into an FCh, thus overcoming communication difficulties between developers, as source code can be read and interpreted in the form of independent, easy-to-understand diagrams [8].

In the research, Ying and Feng [9] describe a low-level FCh-based language that can be used to implement high-level quantum languages and design quantum compilers. The formal semantics of the FCh language was formulated and the notion of the correctness of programs written in that language was introduced.

The literature reviewed above also shows that the use of FChs is of principal importance in the development of cognitive processes, communication and other areas of research as well. However, the disadvantage of FChs is that they can easily become large and difficult to fix. These disadvantages can be overcome with the NS diagram, as the entire algorithm can be represented without edges. Figure 1 shows the same algorithm description with the FCh and the NS diagram. The difference in size is clearly visible.

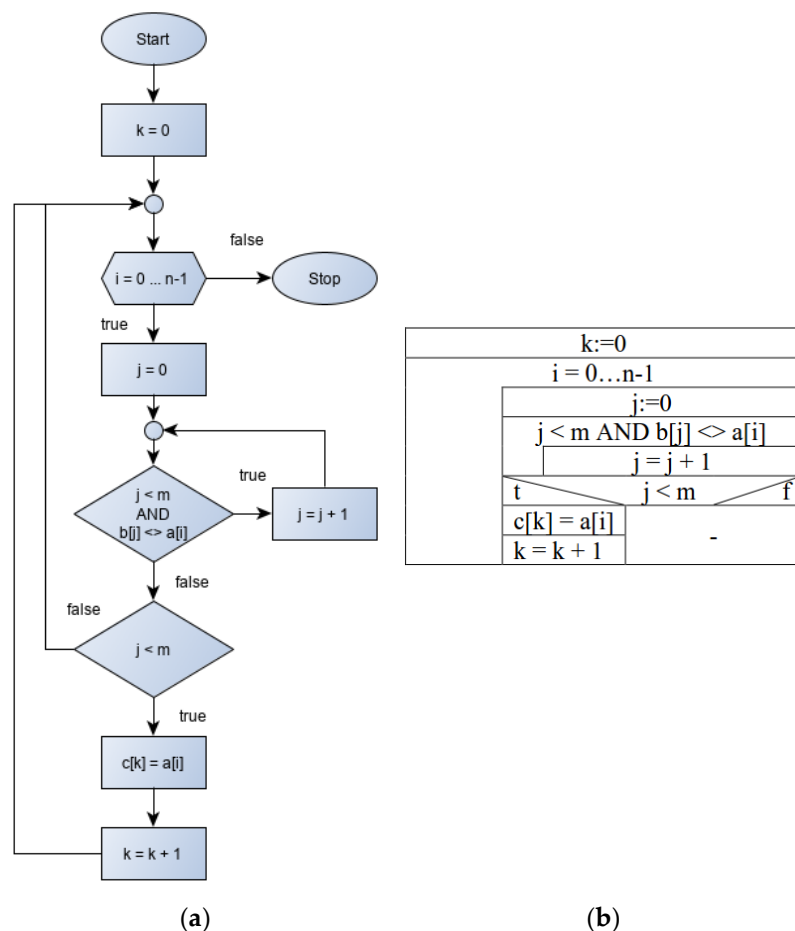


Figure 1. The FCh (a) and the N-S or the structogram diagram (b) of Intersection algorithm.

However, the difference in size does not necessarily mean that the NS diagram is easier to understand. If the intelligibility and readability of the tools describing each algorithm could be measured in an accurate and objective way, it could be of help in selecting the appropriate tool, which would be of great support, especially for beginning programmers, in learning more complex algorithms to create systems that define the technology of the society of the future [10].

Summarizing Charntaweekhun and Wangsiripitak [1], who used the FCh for syntax-independent visual programming, Xinogalos [5] made suggestions for using the FCh for more effective software development education; Cabo [6] and Hooshyar et al. [7] see the FCh as a tool to support the development of cognitive processes; for Shafeek and Karunarathne [8] communication is easier to implement between developers than the source code based sort; Ying and Feng [9] said the FCh can help with high-level quantum programming. However, I examined the intelligibility of the FCh using eye movement parameters and compared it with the intelligibility of the N-S as since these algorithm description tools have not been examined by anyone else in this way. All in all, the article examines that the NS diagram can be read and interpreted with less or more cognitive load, and the implementation of the algorithms can be done with a lower or higher mental load.

## 2. Theoretical Background

There are basically three main approaches to measuring cognitive load [11]. The first method is to analyze performance indicators, such as the total score achieved in a task [12,13], or the time spent on a task [14]. However, the disadvantages of this type of measurement are that performance indicators need to be personalized to the task and monitoring of continuous cognitive force is not feasible either, as scores are obtained at certain intervals or after completion of a task. The second method is the questionnaire approach, which provides information about the difficulty of the task on the basis of subjective measurement; moreover, in this case we get these opinions mostly after the completion of the task [15]. The most reliable technique is based on physiological observation, which allows us to obtain real-time information about a person's cognitive load without having to interrupt the task-solving process [16].

Non-invasive psychophysical tools provide an opportunity for researchers to measure a person's emotional states [17], attention [18] or cognitive load in real time. Recently, affordable and high-resolution eye movement tracking devices have been developed to record eye movement parameters and help the eye-tracking-based research [19]. These devices produce the main eye-tracking parameters such as fixations (relatively stable focus of eyes on the element of the presented object), saccades (quick movements between 30 and 120 milliseconds of the eyes between two fixation points) and pupil diameter [20]. These parameters can provide information about a test subject's behavior, level of visual attention or cognitive status [21]. Conclusions can be drawn from, for example, examining how long a given person looks at a given object on the screen (fixation duration) and how many times (number of fixations) [22]. If someone spends a lot of time in a given position (the duration of the fixation is long), it may indicate that they have difficulty interpreting the information in a given position [20,21,23,24]. In addition, knowing the duration of fixation, we can also create a so-called attention map, which allows further analyses [25]. The results of Just and Luftos [26–28] show that a relationship can be discovered between eye fixations and cognitive load, according to which the more and longer the number and duration of fixations, the greater the cognitive load. In addition to the examination of fixations, pupillometry is widely used to examine cognitive load, since under constant light conditions, the size of the pupil changes systematically to the limit of mental effort [29–31]. For example, at focusing and concentration under constant light conditions, the diameter of the pupil gradually increases [32].

The Area of Interest (AOI) is a distinguished, high-priority area that can be examined separately. Indicators used in AOI analyses, such as the length and number of fixations and the number of visits (returns to the area), were the most commonly used, as they

may also indicate the subjective importance of a particular area [20–23]. Furthermore, the higher the number of visits to the AOI area, the higher the cognitive load on processing the information there [20–23,33].

All in all, the following research questions are addressed in this study:

Research Question 1 (RQ1): Is there a significant difference in the visual parameters (fixation duration mean, number of fixations and average of pupil diameter) in the interpretation of the FCh and the N-S?

Research Question 2 (RQ2): How are visual parameters (fixation duration mean, number of fixations and average of pupil diameter) distributed over implementation when test subjects are using FCh compared to N-S?

### 3. Materials and Methods

In the research, the eye movement parameters of test subjects were recorded using a GazePoint 3 (<https://www.gazept.com/product/gazept-gp3-eye-tracker/> (accessed on 27 December 2021)) eye-tracker tool and the OGAMA (<http://www.ogama.net/> (accessed on 27 December 2021)) open source software package while implementing a randomly selected algorithm on an FCh or N-S diagram in a C# programming language in a Visual Studio development environment.

#### 3.1. Test Subjects

The study involved 42 test subjects (12 woman and 30 man) between the ages of 18 and 22 ( $M = 19.8$ ,  $SD = 1.74$ ) who were not on medication and declared themselves completely healthy, with no psychiatric or neurological disorders and with no difficulties in reading or learning in the past and during the test and who applied for the test on a voluntary basis. Based on the results of the programming subject, those with similarly good, better-than-average programming skills were selected and knew the tools used in the study.

#### 3.2. Test Conditions and Steps of the Research

The GP3 eye-tracker hardware unit was placed about 65 cm from the eye of test subjects. In all cases, I tried to ensure that the illumination of the test subjects was naturally uniform and that no sudden changes in light conditions occurred. I displayed the algorithm descriptions and the source code editing interface of the Visual Studio development environment on a 22" diameter LG22M45 monitor capable of  $1920 \times 1080$ . A schematic diagram of the testing environment is shown in Figure 2.

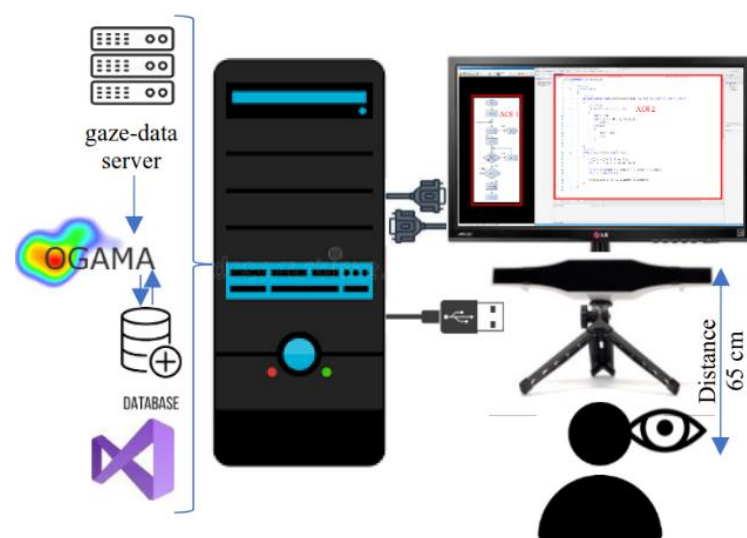


Figure 2. A schematic diagram of equipment setup.

To obtain additional information on eye movement parameters, I marked these areas AOI in OGAMA, as illustrated in Figure 3. On the left side (AOI1) of the display is an image of the algorithm to be implemented represented by a randomly selected algorithm description tool (FCh: AOI1<sub>FCh</sub> or N-S diagram: AOI1<sub>N-S</sub>), while on the right side (AOI2) it was possible to implement the source code (FCh-based implementation: AOI2<sub>FCh-based implementation</sub> and N-S-based implementation: AOI2<sub>N-S-based implementation</sub>). The fixation detection algorithms of OGAMA come from LC Technologies, which is a dispersion-type algorithm with window and was ported to C# and a time estimation support was added. A detailed description of the algorithm can be found in reference [34].

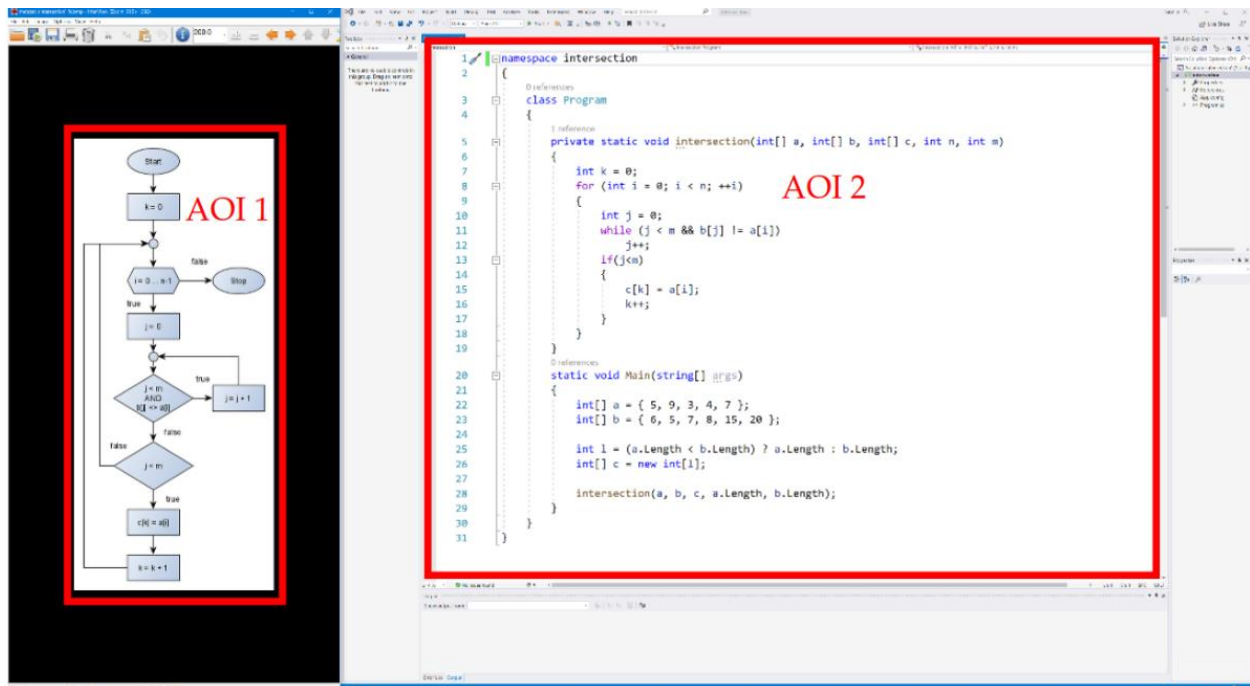
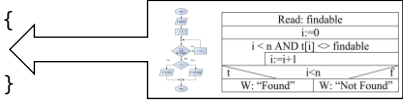


Figure 3. Definition of AOIs.

The algorithm description tools randomly represented one of the algorithms detailed in Section 3.3. In any case, in order to avoid that level of the test subject's knowledge concerning the results of the research, each test subject must also implement an algorithm on 1 FCh and 1 N-S. The density of the figures or texts also affects the processing of information [35], so clarity and easy readability are important factors in the algorithm description tools and program codes; therefore, I tried to keep the same distance between the texts. A total of 42 FChs and 42 N-S-based eye movement parameter packages were saved in a database for further evaluation.

During the implementation, all that was needed was to implement the algorithm, as the additional source code snippet was already available to everyone. Figure 4 illustrates such a snippet of code.

```

namespace IntersectionApp
{
    class Program
    {
        private static void intersect(int[] a, int[] b, int[] c, int n, int m)
        {
            
        }
        static void Main(string[] args)
        {
            int[] a = { 5, 9, 3, 4, 7 };
            int[] b = { 6, 5, 7, 8, 15, 20 };

            int l = (a.Length < b.Length) ? a.Length : b.Length;
            int[] c = new int[l];

            intersect(a, b, c, a.Length, b.Length);
        }
    }
}

```

**Figure 4.** The available source code snippet.

### 3.3. Applied Algorithms

Test subjects had to implement randomly selected algorithms already learned in their previous studies in C# based on an FCh or N-S. During the implementation of the Decision algorithm, it had to be decided whether an element (findable) with a given property could be found in the array ( $T$ ) incoming in the parameter. As soon as the test condition was met, the applicable cycle had to stop as it would have been unnecessary to continue running. If the loop had stopped because we also exceeded the last test element in the array, the element we were looking for could not be found in the array. The pseudo-description of the algorithm is shown in Algorithm 1.

---

#### Algorithm 1: Possible Implementation of Decision

---

- 1: **data:**  $T$ : input array;  $N$ : length of array;  $findable$ : value to look for.
  - 2: **procedure** DECISION ( $T, N, findable$ )
  - 3:   **while**  $i \leq N$  and  $T[i] \neq findable$  **do**
  - 4:      $i \leftarrow i + 1$
  - 5:   **end while**
  - 6:   **if**  $i \leq N$  **then**
  - 7:     **output:** "Found"
  - 8:   **else**
  - 9:     **output:** "Not Found"
  - 10:   **end if**
  - 11: **end procedure**
- 

During the implementation of the Intersection algorithm, the identical elements of two arrays ( $A[1 \dots N]$  és  $B[1 \dots M]$ ) had to be selected into a third, C array. The problem to be solved can only be interpreted accurately if an element is not found twice in each array. The maximum number of elements in array C is the smaller of  $N$  and  $M$ . The pseudo-description of the algorithm is shown in Algorithm 2.

**Algorithm 2: Possible Implementation of Intersection**


---

```

1: data:  $A, B, C$ : input arrays;  $N$ : length of  $A$ ;  $M$ : length of  $B$ 
2: procedure INTERSECTION ( $A, B, C, N, M$ )
3:    $k \leftarrow 0$ 
4:   for  $i \leftarrow 1$  to  $N$  do
5:      $j \leftarrow 0$ 
6:     while  $j \leq M$  and  $B[j] \neq A[i]$  do
7:        $j \leftarrow j + 1$ 
8:     end while
9:     if  $j \leq M$  then
10:       $c[k] \leftarrow a[i]$ 
11:       $k \leftarrow k + 1$ 
12:    end if
13:  end for
14: end procedure

```

---

During the implementation of the Union algorithm, the elements of two arrays ( $A[1 \dots N]$  and  $B[1 \dots M]$ ) that are in at least one of the arrays are placed in a third array  $C$ . The pseudo-description of the algorithm is shown in Algorithm 3.

**Algorithm 3: Possible Implementation of Union**


---

```

1: data:  $A, B, C$ : input arrays;  $N$ : length of  $A$ ;  $M$ : length of  $B$ 
2: procedure UNION ( $A, B, C, N, M$ )
3:   for  $i \leftarrow 1$  to  $N$  do
4:      $c[i] \leftarrow a[i]$ 
5:    $k \leftarrow n$ 
6:   for  $j \leftarrow 1$  to  $M$  do
7:      $i \leftarrow 0$ 
8:     while  $i \leq N$  and  $B[i] \neq A[i]$  do
9:        $i \leftarrow i + 1$ 
10:    end while
11:    if  $i \geq N$  then
12:       $c[k] \leftarrow b[j]$ 
13:       $k \leftarrow k + 1$ 
14:    end if
15:  end for
16: end procedure

```

---

**4. Results**

The results were determined from the eye movement parameters observed and recorded in all AOI areas. In selecting the appropriate statistical tests, it was considered that the test subjects were independent of each other and the same subjects within a group were examined. It was also determined that, for the examination of the distribution of the variables, the Shapiro–Wilk test was applied. In case of the applied statistical tests,  $p < 0.05$  value was determined as significant.

**4.1. Fixation Duration Mean Based on AOIs**

A brief descriptive statistic of the fixation duration mean results in both AOIs is summarized in Tables 1 and 2.

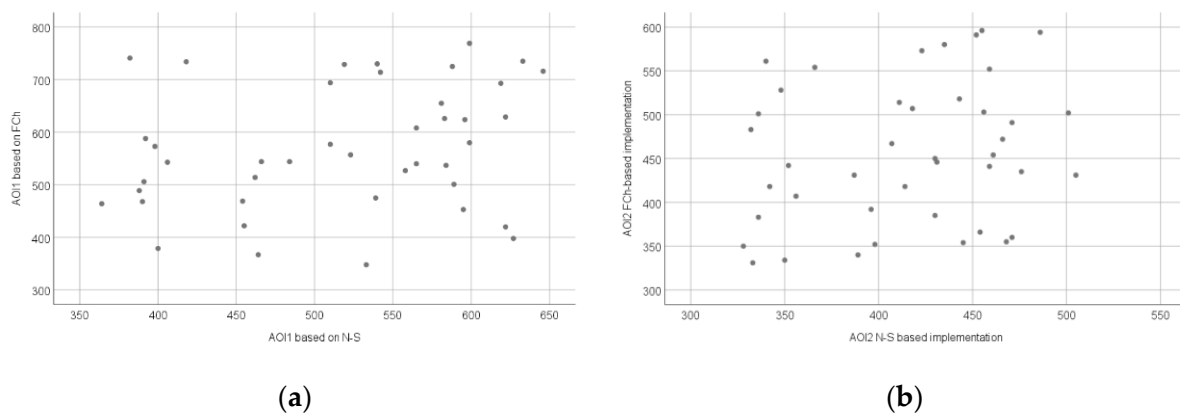
**Table 1.** The descriptive statistic of the fixation duration mean based on AOI1 ( $N = 42$ ).

AOI1 <sub>FCh</sub> (Milliseconds)				AOI1 <sub>N-S</sub> (Milliseconds)			
Min	Max	Mean	SD	Min	Max	Mean	SD
348	769	569.2	117.8	364	646	516.69	86.27

**Table 2.** The descriptive statistic of the fixation duration mean based on AOI2 ( $N = 42$ ).

AOI2 <sub>FCh-based implementation</sub> (Milliseconds)				AOI2 <sub>N-S-based implementation</sub> (Milliseconds)			
Min	Max	Mean	SD	Min	Max	Mean	SD
331	596	456.24	80.46	328	505	414.67	53.27

Figure 5a shows the relationship between the fixation duration mean measured during the interpretation of the FCh and N-S diagrams in the AOI1 area, while Figure 5b shows the relationship between the fixation duration mean measured during the implementation of the source codes based on the FCh and N-S diagrams in the AOI2 area.

**Figure 5.** Scatter plot of the fixation duration mean (milliseconds) based on all AOIs: AOI1 (a) and AOI2 (b).

As a further part of the evaluation, a statistical test was used to determine whether the recorded data differed significantly.

The normality results of the fixation duration mean measured in the AOI1 area are significant (AOI1<sub>FCh</sub>:  $W(42) = 0.957$ ,  $p = 0.111$ , AOI1<sub>N-S</sub>:  $W(42) = 0.920$ ,  $p = 0.006$ ); therefore the Wilcoxon signed-rank test was applied (AOI1:  $T = -242$ ,  $Z = 2.62$ ,  $p = 0.009$  (2-tailed),  $r = 0.286$ ), based on which it can be stated that the fixation duration mean was significantly longer with a small effect in the case in which AOI1<sub>fc</sub>:  $Mdn = 550.5$  milliseconds than in case in which AOI1<sub>N-S</sub>:  $Mdn = 536$  milliseconds.

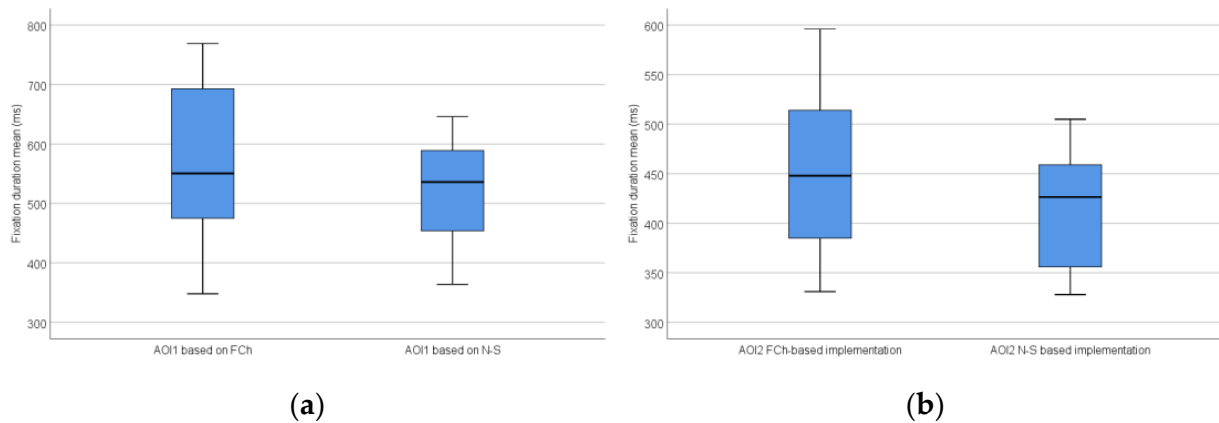
In AOI2 the normality results are significant too (AOI2<sub>FCh-based implementation</sub>:  $W(42) = 0.952$ ,  $p = 0.078$ , AOI2<sub>N-S-based implementation</sub>:  $W(42) = 0.928$ ,  $p = 0.011$ ), so the Wilcoxon signed-rank test was used as well (AOI2:  $T = -231$ ,  $Z = 2.75$ ,  $p = 0.006$  (2-tailed),  $r = 0.3$ ), based on which it can be stated that the fixation duration mean was significantly longer with a medium effect in the case in which AOI2<sub>FCh-based implementation</sub>:  $Mdn = 448$  milliseconds than in the case in which AOI2<sub>N-S-based implementation</sub>:  $M = 426.5$  milliseconds.

In response to the RQ1 question, there was a significant difference in the fixation duration mean ( $p = 0.009$ ) when the test subjects examined and studied different types of algorithm descriptions to be implemented. The results show that in the case of the N-S, the algorithm to be implemented can be interpreted with a shorter fixation result, therefore a lower cognition load, as opposed to the FCh-based description.



In response to the RQ2 question, there was a significant difference in the fixation duration mean ( $p = 0.006$ ) when the test subjects implemented the algorithms to be implemented based on different types of algorithm descriptions. The results show that using the N-S diagram-based algorithm description, the implementation can be performed with a shorter fixation duration mean, so lower cognition load, as opposed to the FCh-based implementation.

The distribution of the fixation duration mean based on all AOIs is shown in Figure 6.



**Figure 6.** The distribution of the fixation duration mean (milliseconds) based on all AOIs: AOI1; (a) and AOI2; (b).

#### 4.2. Number of Fixations Based on AOIs

A brief descriptive statistic of the number of fixations results in both AOIs is summarized in Tables 3 and 4.

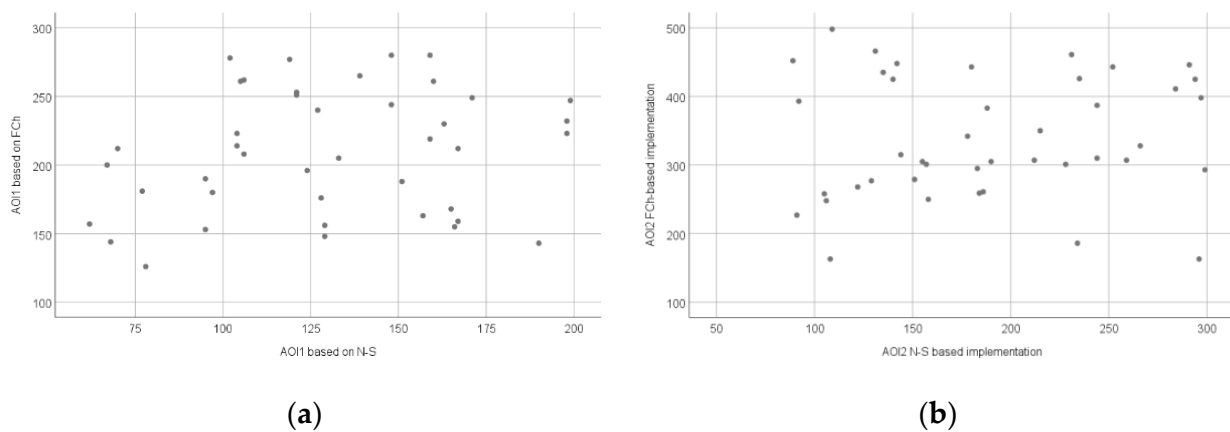
**Table 3.** The descriptive statistic of the number of fixations based on AOI1 ( $N = 42$ ).

AOI1 <sub>FCh</sub> (Count)				AOI1 <sub>N-S</sub> (Count)			
Min	Max	Mean	SD	Min	Max	Mean	SD
126	280	209.7	44.74	62	199	130.29	38.08

**Table 4.** The descriptive statistic of the number of fixations based on AOI2 ( $N = 42$ ).

AOI2 <sub>FCh-based implementation</sub> (Count)				AOI2 <sub>N-S-based implementation</sub> (Count)			
Min	Max	Mean	SD	Min	Max	Mean	SD
163	498	339	88.35	89	299	188.9	65.69

Figure 7a shows the relationship between the number of fixations measured during the interpretation of the FCh and N-S diagrams in the AOI1 area, while Figure 7b shows the relationship between the number of fixations measured during the implementation of the source codes based on the FCh and N-S diagrams in the AOI2 area.

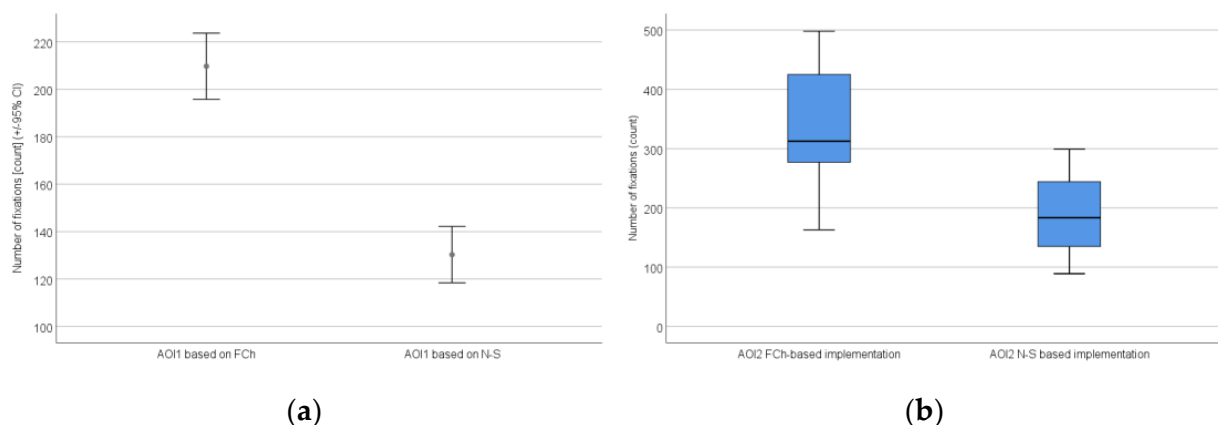


**Figure 7.** Scatter plot of the number of fixations (count) based on all AOIs: AOI1 (a) and AOI2 (b).

As a further part of the evaluation, a statistical test was used to determine whether the recorded data differed significantly. The normality results of the number of fixations measured in the AOI1 area are not significant (AOI1<sub>FCh</sub>:  $W(42) = 0.953$ ,  $p = 0.081$ , AOI1<sub>N-S</sub>:  $W(42) = 0.965$ ,  $p = 0.230$ ), so a paired sample  $t$ -test was used (AOI1:  $t(41) = 9.786$ ,  $p < 0.001$  (2-tailed),  $r = 0.837$ ), based on which it can be stated that the number of fixations was significantly more with a large effect in the case in which AOI1<sub>FCh</sub>:  $M = 209.74$ ,  $SD = 44.74$  than in the case in which AOI1<sub>N-S</sub>:  $M = 130.29$ ,  $SD = 38.08$ .

In AOI2 the normality results are significant (AOI2<sub>FCh-based implementation</sub>:  $W(42) = 0.952$ ,  $p = 0.077$ , AOI2<sub>N-S-based implementation</sub>:  $W(42) = 0.942$ ,  $p = 0.034$ ), so the Wilcoxon signed-rank test was applied (AOI2:  $T = 21.5$ ,  $Z = 5.377$ ,  $p < 0.001$  (2-tailed),  $r = 0.587$ ), based on which it can be stated that the number of fixations was significantly more with a large effect in the case in which AOI2<sub>FCh-based implementation</sub>:  $Mdn = 312.5$  than in the case in which AOI2<sub>N-S-based implementation</sub>:  $Mdn = 183.5$ .

The confidence intervals and the distribution of the number of fixations based on all AOIs are shown in Figure 8.



**Figure 8.** The confidence intervals (a) and the distribution (b) of the number of fixations based on all AOIs: AOI1 (a) and AOI2 (b).

In response to RQ1, there was a significant difference in the number of fixations ( $p < 0.001$ ) when the test subjects examined and studied the different types of algorithm descriptions to be implemented. The results show that in the case of the N-S, the algorithm to be implemented can be interpreted with less fixations, therefore a lower cognition load, as opposed to the FCh-based description.

In response to RQ2, there was a significant difference in the number of fixations ( $p < 0.001$ ) when the test subjects implemented the algorithms to be implemented based

on different types of algorithm descriptions. The results show that, using the N-S-based algorithm description, the implementation can be performed with less fixations, so a lower cognition load, compared to the FCh-based implementation.

### 4.3. Average of Pupil Diameter Based on AOIs

A brief descriptive statistic of the average of pupil diameter results in both AOIs is summarized in Tables 5 and 6.

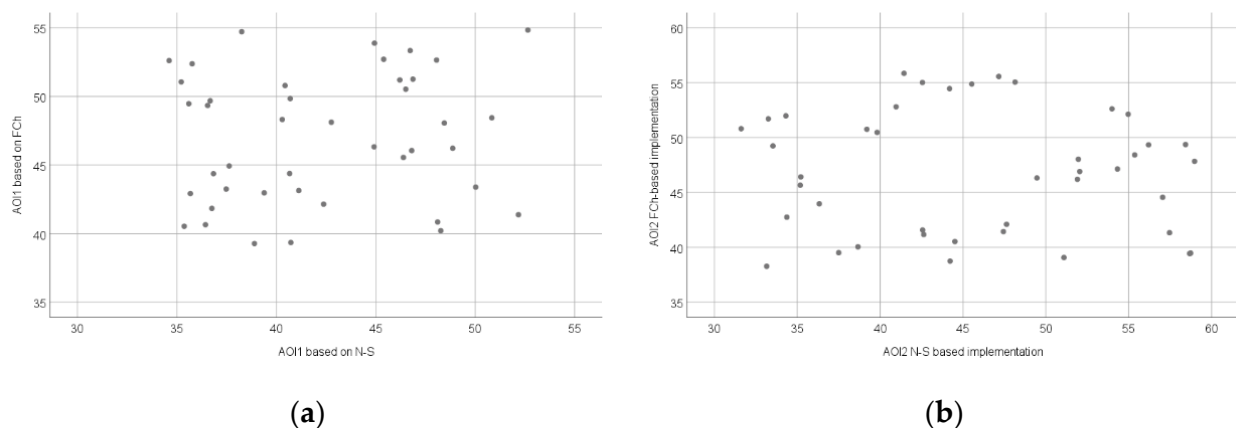
**Table 5.** The descriptive statistic of the average of pupil diameter in AOI1 ( $N = 42$ ).

AOI1 <sub>FCh</sub> (Pixels)				AOI1 <sub>N-S</sub> (Pixels)			
Min	Max	Mean	SD	Min	Max	Mean	SD
39.28	54.84	46.98	4.71	34.61	52.65	42.35	5.41

**Table 6.** The descriptive statistic of the average of pupil diameter in AOI2 ( $N = 42$ ).

AOI2 <sub>FCh-based implementation</sub> (Pixels)				AOI2 <sub>N-S-based implementation</sub> (Pixels)			
Min	Max	Mean	SD	Min	Max	Mean	SD
38.28	55.85	46.88	5.52	31.61	58.97	42.75	8.62

Figure 9a shows the relationship between the average of pupil diameter measured during the interpretation of the FCh and N-S diagrams in the AOI1 area, while Figure 9b shows the relationship between the average of pupil diameter measured during the implementation of the source codes based on the FCh and N-S diagrams in the AOI2 area.



**Figure 9.** Scatter plot of the average of pupil diameter based on all AOIs: AOI1 (a) and AOI2 (b).

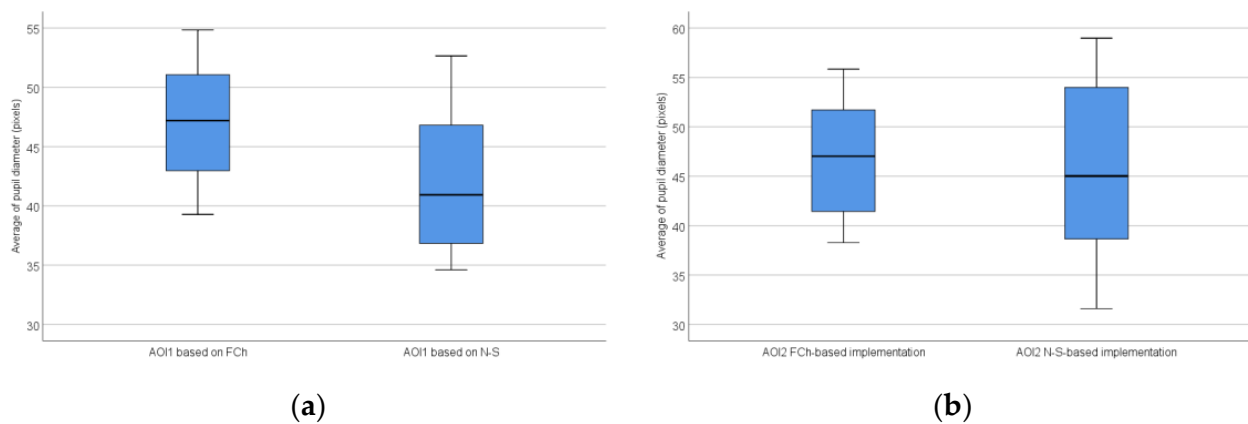
As a further part of the evaluation, a statistical test was used to determine whether the recorded data differed significantly. The normality results of the average of pupil diameter measured in the AOI1 area are significant (AOI1<sub>FCh</sub>:  $W(42) = 0.947$ ,  $p = 0.050$ , AOI1<sub>N-S</sub>:  $W(42) = 0.928$ ,  $p = 0.011$ ); therefore the Wilcoxon signed-rank test was applied (AOI1:  $T = -156$ ,  $Z = 3.695$ ,  $p < 0.001$  (2-tailed),  $r = 0.403$ ), based on which it can be stated that the average of pupil diameter was significantly larger with a medium effect in the case in which AOI1<sub>FCh</sub>:  $Mdn = 47.195$  pixels than in the case in which AOI1<sub>N-S</sub>:  $Mdn = 40.930$  pixels.

In AOI2 the normality results are significant (AOI2<sub>fc based implementation</sub>:  $W(42) = 0.938$ ,  $p = 0.025$ , AOI2<sub>N-S-based implementation</sub>:  $W(42) = 0.940$ ,  $p = 0.028$ ), so the Wilcoxon signed-rank test was used (AOI2:  $T = -391.5$ ,  $Z = 0.750$ ,  $p = 0.453$  (2-tailed),  $r = 0.082$ ), based on which it can be stated that the average of pupil diameter was not significantly larger in the case in which AOI2<sub>FCh-based implementation</sub>:  $Mdn = 47.025$  pixels than in the case in which AOI2<sub>N-S-based implementation</sub>:  $Mdn = 45.020$  pixels.

The distribution of the average of pupil diameter based on all AOIs is shown in Figure 10.

In response to RQ1, the average of pupil diameter showed a significant difference ( $p < 0.001$ ) when the test subjects studied the different types of algorithm descriptions to be implemented. The results show that in the case of N-S, the average of pupil diameter is smaller, so the algorithm to be implemented can be interpreted with a lower cognition load, as opposed to the FCh-based description.

In response to RQ2, in the average of pupil diameter there was not a significant difference ( $p = 0.453$ ) when the test subjects used an N-S or FCh in the implementation of the algorithm to be implemented, so using this eye-tracking parameter it cannot be clearly stated that the implementation can be performed with a different cognition load; however, a kind of tendency can be seen here as well, during which it can be seen that by interpreting the N-S, the implementation can be implemented, if not significantly, with a slightly lower mental load than using the FCh.



**Figure 10.** The distribution of the average of pupil diameter based on all AOIs: AOI1 (a) and AOI2 (b).

## 5. Discussion

After a detailed evaluation of the results of the eye tracking parameters recorded, it can be stated that the N-S can be read and interpreted with less cognitive load, and the implementation of the algorithms can be done with a lower mental load. In the research, the number of fixations, fixation duration means and pupil diameters, which indicate the level of attention and cognitive condition of the test subjects, also suggest this. This is because in the case of N-S, significantly less and shorter information recording and processing were necessary to understand and implement the algorithm.

In addition, after testing, in a short interview, test subjects summarized their experiences and shared what they thought about which algorithm descriptive tool best supported their work. The responses received and the opinions expressed clearly support what was revealed and described in the evaluation of the results. It was said that the length of the FCh and the diversity of the arrows made it difficult to understand how the algorithm works. Using the more concise N-S, for example, they were able to decide much more clearly and in less time how long a loop would last.

The results revealed and defined in the research can greatly contribute to the more efficient completion of the implementation phase of the software development lifecycle at a lower cost. This may be especially true for programming learners and beginner programmers. When developing an application, the right tools can help you teach and learn more efficiently and effectively. It can be stated that in order to increase the efficiency of knowledge transfer and learning in the field of education, it is worth introducing the N-S from the beginning of programming teaching, thus making the understanding of an algorithm more efficient with lower cognitive load, avoiding difficulties and problems occurring from the understanding and readability of the syntax. In addition to reducing the costs of each development phase in the industry, we can create a source code that is

easier to read and maintain if we always choose the right tool or programming technique. Devices that record the route of such a gaze can help in this. Apparently, testing may seem like a waste of time at the beginning of the software development phase, but it can pay off later, even in the implementation phase, and further phases such as testing and evolution can be significantly decreased.

In addition, the reliability and accuracy of hardware and software devices that record eye movement parameters are improving, and their costs are decreasing. They can facilitate the selection of personalized or team-specific support tools, programming techniques and technologies based on recorded and evaluated eye movement results. This improves the collaboration and communication of development teams and further optimizes their coding styles. Eventually, the evolution of software, its upgradeability and its maintainability can be significantly improved, and their costs can be decreased.

In the future, with the applied eye movement tracking system, additional programming tools, techniques and technologies may become available for cognitive load. For students in education, and for industry development teams in the industry, the use of tools, methods, techniques and technologies that require less mental effort can result in a more effective learning and development process. In further research, using the developed eye movement tracking system, I would like to investigate additional programming tools, methods, solutions, techniques and technologies in terms of mental load in relation to eye movement parameters.

## 6. Conclusions

Research based on eye movement tracking is expected to emerge in the near future for the study of cognitive processes such as programming. The study examined the readability and interpretability of two algorithm description tools: The FCh and the N-S involving test subjects. In the study, an eye-tracking-based test system was developed, with the help of which the gaze path of the test subjects can be observed, recorded and evaluated, and then conclusions can be drawn after determining the obtained results. The article explores the relationships between the cognitive load and eye movement parameters of test subjects and the results revealed using the test system, and emphasizes the importance and significance of properly selected tools, programming techniques and technologies.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The author would like to acknowledge the financial support of Emberi Erőforrások Minisztériuma (EMMI) under the NTP-NFTÖ-21-B-0058.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Charntaweekhun, K.; Wangsiripitak, S. Visual Programming Using Flowchart. In Proceedings of the 2006 International Symposium on Communications and Information Technologies, Bangkok, Thailand, 18–20 October 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 1062–1065.
2. Kovari, A. Study of Algorithmic Problem-Solving and Executive Function. *Acta Polytech. Hung.* **2020**, *17*, 241–256. [[CrossRef](#)]
3. Francisti, J.; Balogh, Z.; Reichel, J.; Magdin, M.; Koprda, S.; Molnár, G. Application Experiences Using IoT Devices in Education. *Appl. Sci.* **2020**, *10*, 7286. [[CrossRef](#)]
4. Kovari, A.; Rajcsányi-Molnár, M. Mathability and Creative Problem Solving in the MaTech Math Competition. *Acta Polytech. Hung.* **2020**, *17*, 147–161. [[CrossRef](#)]
5. Xinogalos, S. Using Flowchart-Based Programming Environments for Simplifying Programming and Software Engineering Processes. In Proceedings of the 2013 IEEE Global Engineering Education Conference (EDUCON), Berlin, Germany, 13–15 March 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1313–1322.
6. Cabo, C. Effectiveness of Flowcharting as a Scaffolding Tool to Learn Python. In Proceedings of the 2018 IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA, 3–6 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–7.

7. Hooshyar, D.; Ahmad, R.B.; Nasir, M.H.N.M.; Mun, W.C. Flowchart-Based Approach to Aid Novice Programmers: A Novel Framework. In Proceedings of the 2014 International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 3–5 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–5.
8. Shafeek, N.; Karunaratne, D.D. A Prototype Compiler to Convert Source-Code to Flowchart. In Proceedings of the 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, 26–29 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 157–167.
9. Ying, M.; Feng, Y. A Flowchart Language for Quantum Programming. *IEEE Trans. Softw. Eng.* **2010**, *37*, 466–485. [[CrossRef](#)]
10. Kovari, A. The synergy of digital society and digital education. *Civ. Szle.* **2020**, *17*, 69–72.
11. Charleton, S.; O'Brien, T. Measurement of Cognitive States in Testing and Evaluation. In *Handbook of Human Factors and Evaluation*; CRC Press: Boca Raton, FL, USA, 2002; pp. 97–126.
12. Guzsvinecz, T.; Orbán-Mihálykó, É.; Perge, E.; Sik-Lányi, C. Analyzing the spatial skills of university students with a Virtual Reality application using a desktop display and the Gear VR. *Acta Polytech. Hung.* **2020**, *17*, 35–56. [[CrossRef](#)]
13. Guzsvinecz, T.; Sik-Lányi, C.; Orbán-Mihálykó, É.; Perge, E. The Influence of Display Parameters and Display Devices over Spatial Ability Test Answers in Virtual Reality Environments. *Appl. Sci.* **2020**, *10*, 526. [[CrossRef](#)]
14. Guzsvinecz, T.; Orbán-Mihálykó, É.; Sik-Lányi, C.; Perge, E. Investigation of spatial ability test completion times in virtual reality using a desktop display and the Gear VR. *Virtual Real.* **2021**, 1–14. [[CrossRef](#)]
15. Hart, S.G.; Staveland, L.E. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Advances in Psychology*; Elsevier: Amsterdam, The Netherlands, 1988; Volume 52, pp. 139–183.
16. Card, S.; Moran, T.; Newell, A. The Model Human Processor- An Engineering Model of Human Performance. In *Handbook of Perception and Human Performance*; Wiley-Interscience: Hoboken, NJ, USA, 1986; p. 2.
17. Magdin, M.; Balogh, Z.; Reichel, J.; Koprda, Š.; György, M. Automatic detection and classification of emotional states in virtual reality and standard environments (LCD): Comparing valence and arousal of induced emotions. *Virtual Real.* **2021**, *25*, 1029–1041. [[CrossRef](#)]
18. Kovari, A.; Katona, J.; Costescu, C. Quantitative Analysis of Relationship between Visual Attention and Eye-Hand Coordination. *Acta Polytech. Hung.* **2020**, *17*, 77–95. [[CrossRef](#)]
19. Kovari, A.; Katona, J.; Costescu, C. Evaluation of Eye-Movement Metrics in a Software Debugging Task using GP3 Eye Tracker. *Acta Polytech. Hung.* **2020**, *17*, 57–76. [[CrossRef](#)]
20. Holmqvist, K.; Nyström, M.; Andersson, R.; Dewhurst, R.; Jarodzka, H.; Van de Weijer, J. *Eye Tracking: A Comprehensive Guide to Methods and Measures*; OUP Oxford: Oxford, UK, 2011.
21. Derick, L.-R.; Gabriel, G.-S.; Máximo, L.-S.; Olivia, F.-D.; Noé, C.-S.; Juan, O.-R. Study of the User's Eye Tracking to Analyze the Blinking Behavior While Playing a Video Game to Identify Cognitive Load Levels. In Proceedings of the 2020 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, Mexico, 4–6 November 2020; IEEE: Piscataway, NJ, USA, 2020; Volume 4, pp. 1–5.
22. Tsai, M.-J.; Hou, H.-T.; Lai, M.-L.; Liu, W.-Y.; Yang, F.-Y. Visual Attention for Solving Multiple-Choice Science Problem: An Eye-Tracking Analysis. *Comput. Educ.* **2012**, *58*, 375–385. [[CrossRef](#)]
23. Evinger, C.; Manning, K.A.; Sibony, P.A. Eyelid Movements. Mechanisms and Normal Data. *Investig. Ophthalmol. Vis. Sci.* **1991**, *32*, 387–400.
24. Orchard, L.N.; Stern, J.A. Blinks as an Index of Cognitive Activity during Reading. *Integr. Physiol. Behav. Sci.* **1991**, *26*, 108–116. [[CrossRef](#)] [[PubMed](#)]
25. Rakoczi, G.; Pohl, M. Visualisation and Analysis of Multiuser Gaze Data: Eye Tracking Usability Studies in the Special Context of e-Learning. In Proceedings of the 2012 IEEE 12th International Conference on Advanced Learning Technologies, Rome, Italy, 4–6 July 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 738–739.
26. Just, M.A.; Carpenter, P.A. A Theory of Reading: From Eye Fixations to Comprehension. *Psychol. Rev.* **1980**, *87*, 329. [[CrossRef](#)] [[PubMed](#)]
27. Just, M.A.; Carpenter, P.A. Eye Fixations and Cognitive Processes. *Cogn. Psychol.* **1976**, *8*, 441–480. [[CrossRef](#)]
28. Loftus, G.R.; Mackworth, N.H. Cognitive Determinants of Fixation Location during Picture Viewing. *J. Exp. Psychol. Hum. Percept. Perform.* **1978**, *4*, 565. [[CrossRef](#)]
29. Goldwater, B.C. Psychological Significance of Pupillary Movements. *Psychol. Bull.* **1972**, *77*, 340. [[CrossRef](#)]
30. Suzuki, Y.; Hirayama, K.; Shimomura, T.; Uchiyama, M.; Fujii, H.; Mori, E.; Nishio, Y.; Iizuka, O.; Inoue, R.; Otsuki, M.; et al. Changes in Pupil Diameter Are Correlated with the Occurrence of Pareidolias in Patients with Dementia with Lewy Bodies. *NeuroReport* **2017**, *28*, 187. [[CrossRef](#)]
31. Morad, Y.; Lemberg, H.; Yofe, N.; Dagan, Y. Pupillography as an Objective Indicator of Fatigue. *Curr. Eye Res.* **2000**, *21*, 535–542. [[CrossRef](#)]
32. Tsai, Y.-F.; Viirre, E.; Strychacz, C.; Chase, B.; Jung, T.-P. Task Performance and Eye Activity: Predicting Behavior Relating to Cognitive Workload. *Aviat. Space Environ. Med.* **2007**, *78*, B176–B185. [[PubMed](#)]
33. Allsop, J.; Gray, R.; Bulthoff, H.H.; Chuang, L. Effects of Anxiety and Cognitive Load on Instrument Scanning Behavior in a Flight Simulation. In Proceedings of the 2016 IEEE Second Workshop on Eye Tracking and Visualization (ETVIS), Baltimore, MD, USA, 23 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 55–59.

- 
34. Voßkühler, A. OGAMA Description (for Version 2.5). Available online: <http://www.ogama.net/sites/default/files/pdf/OGAMA-DescriptionV25.pdf> (accessed on 14 January 2022).
  35. Shasteen, J.R.; Sasson, N.J.; Pinkham, A.E. Eye Tracking the Face in the Crowd Task: Why Are Angry Faces Found More Quickly? *PLoS ONE* **2014**, *9*, e93914. [[CrossRef](#)] [[PubMed](#)]