Effects of Variable Naming on Computer Program Reading and Comprehension

Mitchell M. Devenport*
Clemson University

Abstract

In this paper, descriptive and non-descriptive variable naming conventions are discussed in the context of how they affect a reader’s ability to comprehend a computer program and how they affect experienced and inexperienced programmers differently. This is explored in depth both quantitatively and qualitatively by analyzing the number of, duration of, and distance between fixations while reading a computer program; the time required to understand a computer program; and a reader’s ability to describe a program functionally and algorithmically after reading it.

CR Categories: H.1.2 [Information Systems]: User/Machine Systems—Human information processing;

Keywords: Eye Tracking, Program Reading, Variable Naming, Obfuscation

1 Introduction

1.1 Motivation

No matter what variable naming convention used, it is most always advised that programmers use descriptive variable names [Kochan 1988]. This study is motivated to determine the actual effect of non-descriptive variable names and how much we actually save later on for ourselves and others by spending time now to carefully and meticulously name our variables.

1.2 Goals

This study has three main objectives:

1. Determine if non-descriptive variable names hinder our ability to comprehend a computer program.

2. Provide evidence of whether the effect is larger on experienced programmers or inexperienced programmers.

3. Create the results necessary for further research on how humans read computer programs and how we can write more easily understood computer programs.

The first two objectives of this study are evaluated both quantitatively and qualitatively. The quantitative assessment was achieved through the use of various statistics obtained through an eye-tracker including: number of fixations, length of fixations, distance between fixations, and the time taken to read a program. The results were assessed qualitatively by asking participants to describe the functionality of a program and then assessing whether or not their description was correct.

1.3 Hypothesis

When compared to programs which use descriptive variable names, programs which use non-descriptive variable names will take longer to read, are less likely to be understood functionally and algorithmically by a reader, and will lead to more fixations with greater distance between subsequent fixations. We also hypothesize that non-descriptive variable names will cause experienced programmers to have more fixations with greater distance between subsequent fixations than inexperienced programmers but will hinder the ability of inexperienced programmers to understand the program more than the experienced programmers.

2 Background

A fair amount of eye tracking studies have been conducted to assess how humans read text. Unfortunately, less research has been conducted on how humans read computer programs. It has previously been noted that programmer’s often read code by quickly scanning the entire program and then focusing most of their attention on the beginning [Bednarik and Randolph 2008]. This is very relevant to a study about the effects of variable naming as it is a common style to place variable names at the top of the program, suggesting that non-descriptive variable names may have a large effect.

3 Methodology

3.1 Apparatus

The study was conducted using a standard desktop computer with a 22” LCD monitor which participants were seated approximately 30 inches in front of. Participant’s eye movements were recorded using the GazePoint GP3 Desktop Eye-Tracker (Figure 1). The GP3 has an accuracy of 0.5 - 1 degree of visual angle and a sampling rate of 60Hz, as reported by the manufacturer.

3.2 Stimulus

Two different, but functionally equivalent, computer programs were presented to each participant: one containing descriptive variable names and the other containing non-descriptive variable names (Figure 2 and Figure 3).

---

* e-mail: mdevenp@clemson.edu
3.3 Subjects

The experiment consisted of 7 experienced programmers and 4 inexperienced programmers between the ages of 17 and 26.

3.4 Experimental Design

The experiment consisted of a 2x2 factorial experimental design with two groups of participants reading two different computer programs. The experiment was within subjects. There were two groups of participants, one consisting of experienced programmers and the other consisting of inexperienced programmers. Both groups were presented with two programs that are functionally equivalent. However, one program contained non-descriptive variable names, while the other contained descriptive variable names. Half of the participants in each group were shown the program with descriptive variables first, while the other half were shown the program with non-descriptive variables first. It was determined prior to the experiment which participants were to be presented which program first.

3.5 Procedure

Participants were asked to participate in an approximately 20 minute long study. Before the study began, participants were informed of the procedure and presented with necessary IRB forms.

3.5.1 Pre-Experiment Survey

Each participant was asked to complete a short survey consisting of questions used to determine their programming proficiency.

3.5.2 Calibration

Each participant was asked to undergo a 10 second long, 5-point calibration process. First a dot was placed in the middle of the screen, then the top-right, bottom-right, bottom-left, and top-left corners. Each dot appeared for several seconds as the participant fixed their gaze on the dot and the eye-tracker calibrated itself.

3.5.3 Read Program 1

Each participant was asked to read the first program for comprehension, in other words they should not stop reading until they fully understand what the program does or feel that they cannot understand it any better. For half of the participants this was the program with descriptive variable names, for the other half it was the program with non-descriptive variable names.

3.5.4 Post-Reading Survey

Each participant was then asked to complete a single question survey on the functionality of the program they had just read to qualitatively determine their functional understanding of the program.
3.5.5 Read Program 2

Each participant was then asked to read the second version of the program for comprehension. This time each participant was given the program that they were not given the first time.

3.5.6 Post-Reading Survey

Each participant was then asked to complete a single question survey on the functionality of the program they had just read to qualitatively determine their functional understanding of the program.

4 Results

Quantitative data was collected for each participant reading each program using the GazePoint Analysis and GazePoint control software and was then exported for further analysis. Fixations and their locations were calculated by a proprietary algorithm used in the GazePoint Analysis software. Qualitative data was also collected from post-reading surveys which was manually compiled and analyzed. These data sets were then used to measure the following six independent variables:

1. Difference in average number of fixations between the program with descriptive and the program with non-descriptive variables.
2. Difference in average number of fixations between groups.
3. Difference in average screen distance separating fixations between the program with descriptive and the program with non-descriptive variables.
4. Difference in average screen distance separating fixations between groups.
5. Difference in average time taken to read and understand the computer program between the program with descriptive variables and the program with non-descriptive variables.
6. Proportion of participants able to describe the functionality of the computer program.
7. Proportion of participants able to describe the algorithm of the computer program.

4.1 Difference in average number of fixations between the program with descriptive and the program with non-descriptive variables

While participants were reading each computer program, their fixations were recorded using proprietary software and algorithms. These fixations were then exported and counted using a computer program written in Python. These numbers were averaged for each computer program read by participants. The results (Figure 5) showed a significant difference between the average number of fixations users had while reading the program with descriptive variables and the program with non-descriptive variables. Participants averaged approximately 319 less fixations while reading the program with descriptive variables.

4.2 Difference in average number of fixations between groups

A quantitative analysis of the average number of fixations of each group while reading both the descriptive variable and non-descriptive variable programs was conducted. While the results (Figure 6) were not significant, it was shown that experienced programmers averaged more fixations when working with non-descriptive variables and inexperienced programmers averaged more fixations when reading a program with descriptive variable names.
Fixation locations were recorded using the x and y-coordinates of the fixation as a fraction of the screen size. In this coordinate system (0,0) was the top left corner of the screen, (0.5,0.5) corresponded to the screen center, and (1,1) indicated the bottom right of the screen.

Fixation data from every participant reading each computer program was used to find the distance between each fixation and the fixation before it. These differences were then averaged together to produce an average distance between fixations for a single participant. The averages for all participants were then averaged resulting in an average distance between fixations for the program with descriptive variable names and the program with non-descriptive variable names.

The results (Figure 7) showed a significant difference between the average distance separating fixations for the program with descriptive variables and the program with non-descriptive variables. While reading the program using descriptive variables, users had an average of 0.04 less fraction of the screen between fixations.

4.3 Difference in average screen distance separating fixations between the program with descriptive and the program with non-descriptive variables

4.4 Difference in average screen distance separating fixations between groups

The distances between fixations for each group reading each computer program were also averaged together. The results (Figure 9 and Figure 10) show a significant difference between the distance separating fixations for experienced and inexperienced programmers for both the program with descriptive variables and the program with non-descriptive variables. It is notable that inexperienced programmers averaged approximately 0.07 fraction of the screen size less between each fixation for the program with descriptive variables, while experienced programmers averaged approximately 0.05 fraction of the screen size less between each fixation for the program with non-descriptive variables.

4.5 Difference in average time taken to read and understand the computer program between the program with descriptive variables and the program with non-descriptive variables

Each participant was timed while they read each computer program. These times were averaged together to produce an average time spent reading the computer program with descriptive variables and the program without descriptive variables. The results (Figure 11) show a significant difference between the two. Participants, on average, spent approximately 237 less seconds reading the program with descriptive variables than the program without descriptive variables.

4.6 Proportion of participants able to describe functionality

The first qualitative measure of the study assessed the ability of participants to understand and therefore describe the functionality of
Figure 11: Average time taken to read and understand the computer program.

Figure 12: Proportion of participants able to describe the functional purpose of the given computer program.

Figure 13: Proportion of participants able to describe the algorithm used to achieve the functional purpose of the given computer program.

4.7 Proportion of participants able to describe the algorithm

The second and final qualitative measure analyzed the ability of participants to understand and therefore describe the algorithm used to achieve the functionality of the program. Results (Figure 13) show that while all experienced programmers were able to determine the correct algorithm when presented with the program containing descriptive variables, only 75% of inexperienced programmers were able to do the same. When asked to read the program containing non-descriptive variables, once again no inexperienced programmers were able to understand the algorithm, while approximately 43% of experienced programmers could.

5 Discussion

This study has shown strong evidence for our hypotheses regarding the differences between reading a computer program with descriptive variables and a program with non-descriptive variables. We found that the average time taken to read a program with descriptive variables was significantly less than the time taken to read a program with non-descriptive variables. This finding supports our hypothesis that programs with non-descriptive variable names will take longer to read on average. We found, through qualitative assessments, that users were less likely to understand the program with non-descriptive variable names both functionally and algorithmically, thus providing evidence for our hypothesis that programs with non-descriptive variables are harder for readers to understand. The study showed that when reading computer programs with non-descriptive variables users had significantly more fixations and that those fixations had, on average, significantly more distance between them. This result supports our hypothesis that programs with non-descriptive variable names result in more fixations with greater distance between subsequent fixations.

Our experiments also showed partial support for our hypothesis that non-descriptive variable names cause experienced programmers to have more fixations with greater distance between subsequent fixations than inexperienced programmers but will hinder the ability of inexperienced programmers to understand the program more than experienced programmers. We found that all inexperienced participants were able to comprehend the functionality of the descriptive program and 75% were able to correctly describe the algorithm of the descriptive program while no inexperienced participants could accurately describe the non-descriptive program functionally or algorithmically. It is also the case that all experienced participants accurately described the descriptive program functionally and algorithmically while approximately 57% accurately described the non-descriptive program functionally and about 47% described the algorithm. These two findings support our hypothesis that the non-descriptive variable names hinder the ability of inexperienced programmers to understand the program. Contrary to our hypothesis, we found that experienced programmers had significantly less distance between fixations while reading the program with non-descriptive variable names and had significantly more distance between fixations when reading the program with descriptive variable names. It is also noteworthy that we did not find a significant difference between the number of fixations for each group for descriptive or non-descriptive variables.

6 Conclusion

This study has shown strong evidence for the benefits of using descriptive variable names in computer programs. However, this paper serves more as a starting point for future research in eye tracking to determine more appropriate methods and conventions for writing computer programs. Further research would bring a greater understanding of how humans comprehend computer programs and
would thus enable us to write more comprehensible computer programs.

References
