IS BIGGER BETTER?
THE EFFECTS OF DISPLAY SIZE ON PROGRAM READING

J. F. REISEL and B. SHNEIDERMAN
Department of Computer Science, University of Maryland, College Park, Maryland 20742.

ABSTRACT
An experiment was conducted in which four window sizes (10, 22, 60, and 120 lines) were compared for their effectiveness for program reading. The largest window, in which the entire program could be seen at one time, was found significantly better, in both time to complete the reading task and the number of correct answers in the reading task. Subjects also preferred the larger windows to the smaller windows.

1. Introduction
The traditional computer screen is small. Measuring less than twelve inches diagonally, it holds no more than 24 lines of 80 characters. This modest size has been proposed as one of the reasons why reading from computer screens has been less satisfactory than reading from paper. However, new technology has permitted the development of larger screens capable of holding many windows equal in size to the traditional screen or one window much larger than the traditional size. Examples of such screens include the Apollo or Sun workstation and the IBM 3290 Plasma Display. If programmers had bigger screens or multiple windows, would they be able to do their tasks, such as comprehension and debugging, faster or more accurately? Would they feel more satisfied? How much of the difference between screen and paper is due to the limited number of lines available on the screen?

Previous research on computer terminal screens has focused on two main areas: formatted displays and text reading. Formatted displays are defined as displays in which the display is selectively searched for information rather than read continuously. Formatted displays differ from programs in several ways. First, designers of formatted displays can lay them out to enhance readability, an option not necessarily available to the programmer, whose programs must adhere to a specified syntax. A related difference is that formatted displays require little interpretation while programs must be studied carefully to be understood. That means that the users of formatted displays do not have to do as much searching for information as do programmers. Finally, formatted displays are often meant to be used by a heterogeneous audience, while programs are generally read only by programmers, an audience with more skills and knowledge.

The second area of relevant research has been in the reading of natural-language text on a computer display. Again, text differs from programs in several ways. First, text is
free-format, in that every line is filled to capacity; sentences follow one after the other. Most programs are not written that way; each statement starts on a separate line of the screen; and, any space at the end of a line after a statement is left blank. Second, text is usually read serially; most readers start at the beginning and continue straight through to the end. It is fairly easy to understand text by reading that way. However, most programs are read in a series of jumps. The reader reads the data declarations, then skips to the main program, then jumps around from procedure to procedure as needed. Third, text is composed mainly of alphabetic characters, while programs have many special symbols and operators with special meanings that must be understood.

Haas and Hayes [1985] found that text reordering could be done significantly faster on a 50-line screen than on a 24-line screen, and that reading comprehension scores were higher on a 50-line screen than on a 24-line screen. They found no difference between screen sizes when the task was proofreading. However, their 50-line screen had more differences with the 24-line screen than just the size. Duchnicky and Kolers [1983], studying 1-line, 2-line, 3-line, 4-line, and 20-line screens, found that text on the 2-line and 20-line screens was read faster than text on the 1-line and 2-line screens. No other comparisons were found significant. Based on these two studies, we can roughly group these screen sizes as 1-2, 3, 4-20-24, and 50, in increasing order of suitability for reading text.

Finally, a small body of experimental results has evolved that actually examines the specific issue of reading programs when more lines are available. Weldon, Norman, and Shneiderman [1986] found no difference in program comprehension between 24-line and 48-line displays when the 48-line display was split between 2 24-line screens. It should be pointed out that this experiment was not designed to study the issue of larger display sizes directly, and their lack of results may be due to the fact that their experiment did not focus on the relevant factors.

The first study to look specifically at the issue of program reading on larger screens was performed by Don Preuss and Phil Shafer at the University of Maryland in 1984. They used the IBM 3290 Plasma Display in three configurations: one window of 22 lines, a second window of 60 lines, and a third window of 120 lines. They tested 21 novice computer users (7 per cell) on program comprehension by seeing how many questions could be correctly answered in a fixed time. They used two programs, both of which required the user to scroll the program regardless of the window size.

Their results found no significant difference between the window sizes. They attributed their finding to two facts: the familiarity of the 22-line window to the novice users and the clutter on the screen caused by the larger window sizes. However, there were other factors that may explain their result. First, their experiment fixed the total time for each subject and counted the number of correct answers to a set of questions. Their questions contained very difficult questions intermixed with very easy questions. There was no way for a subject who got stuck on one difficult question to go on to an easier one. These two
conditions made it quite likely that their subjects would get stuck early in the experiment, with no way to advance to the next question. Since the scores for the experiment were extremely low, it is reasonable to assume that that is what happened in many cases. A related concern is that the users were novice computer science students. It is quite possible that the programs being studied and the questions being asked were too difficult for the novice users to answer. A third factor is that there were only seven subjects per cell.

We learned a great deal about designing such experiments from Preuss and Shafer's work, and decided to look again at the question of reading programs on larger screens.

2. Materials

The IBM 3290 Plasma Display terminal has a display screen which measures 11.5 inches by 14.5 inches, and is capable of holding 62 lines of 160 characters. Two font sizes are available, which change the number of lines and the number of characters per line that can be seen. Also, provision is made for dividing the screen into up to four independent windows, the sizes of which may also vary.

For this experiment, four screen configurations were developed. The first consisted of two windows of 62 lines by 80 characters. The first sixty lines being viewed were displayed in the left window, and the next sixty lines were displayed in the right window. This configuration displayed 120 lines of program text at one time. The second consisted of one 62-line by 80-character window centered in the screen, which displayed 60 lines at one time. The third was a 24-line by 80-character window centered both horizontally and vertically. This configuration, the traditional size of computer screens, displayed 22 lines at one time. The fourth was a 12-line by 80-character window, centered both horizontally and vertically, which displayed 10 lines at one time. (In each display, the lines not used for text are used for title and command lines.)

We used the same font size for each display with a 6 by 12 dot matrix for each character. That is the only way we could implement the 120-line display, and we wanted to keep the font constant so that any effects would be due to the number of lines visible. In the cases of the 22-line and 10-line displays, this meant that there was an large amount of blank space surrounding the text; and, we believe that increasing the font size in the smaller displays might improve performance of people using them.

3. Preliminary Study

Our preliminary study looked at the same three displays examined by Preuss and Shafer. These were the 120-line, the 60-line, and the 22-line displays. Responding to the observation that the questions of Preuss and Shafer were of varying levels of difficulty and measured different levels of expertise, we divided the task of program comprehension into two separate levels, following the syntactic/semantic model of Shneiderman [1980]. On the syntactic level, we devised "program reading" questions, which required the subjects to answer questions merely by examining the code itself. An example of such a question would be, "In which procedure is the variable 'Distance' declared?" These questions are
syntactic in that the subject did not need to be able to understand the program in order to answer them. On the semantic level, we devised "program comprehension" questions, which required the subjects to perform a trace of the program to answer the questions. An example of this type of question would be, "Given this input, what would the output of the program be?" These questions are semantic in that the subject needed to know not only what the individual statements meant, but also how the program as a whole worked.

The subjects for this study were twelve junior or senior undergraduate computer science students who were taking a senior-level course in data structures. All subjects had at least four semesters of computer science classes. Each subject answered a set of reading questions and a set of comprehension questions about a program. The program was a 118-line implementation of Dijkstra's Algorithm for finding the shortest path between all pairs of points on a graph. The program was short enough to fit entirely in the 120-line window. Most of the functions of the IBM XEdit editor were disabled, allowing the subjects to use only the page-up and page-down commands. This was done primarily to prohibit subjects from minimizing the negative effects of a smaller window by using the Locate command to find a string.

There was no significant difference found in any of the dependent variables among the screen sizes. In the case of the correctness measure for the reading task, we speculated that that was due to the nature of the task. The questions were such that everybody did well, regardless of the window condition. In the cases of the other three dependent variables, we speculated that that was due to the combination of the lack of repeated measures and the well-established phenomenon of individual variability, that is, low statistical power. The differences in individual ability so expand the range of performance that it becomes impossible to isolate the effects of the independent variables; and the absence of repeated measures prevents controlling variability by comparing the same subject's performance under different conditions. equally to all conditions. We found ranges of 3.5:1 and 4:1 in the time measures, which confirms the earlier study of Curtis [1981].

4. Main Experiment

Because of the difficulty in the preliminary study caused by programmer variability, we decided to use a repeated-measures design for the second experiment. When each subject did both the reading questions and the comprehension questions, the time per subject exceeded one hour, with the reading questions taking fifteen minutes and the comprehension questions taking over forty-five minutes. By going to a repeated-measures design, we would be increasing the time per subject to over three hours. We decided to focus on reading questions. Since the reading questions would take only about fifteen minutes for a subject to answer, we used this opportunity to add a fourth display, the 10-line display.

With our new design being a four-level, repeated measures experiment, we needed to have four programs. We also wanted to test the effects of being able to see the entire program on the screen at one time, so we decided to have all the programs be less than 120 lines long. These programs were:
(1) a 118-line implementation of Dijkstra's Algorithm for finding the shortest path between all pairs of nodes on a graph (the same program used in the preliminary study);
(2) a 114-line simulation of a vision system looking at planar surfaces at random angles;
(3) a 118-line implementation of Conway's Game of Life;
(4) a 116-line implementation of an electoral system using a transferable-votes scheme.

Not all reading questions are alike. Different skills needed to answer reading questions were identified, and it was decided to use four types of questions. These question types required the subject to locate:

(1) All occurrences of a specified feature, which could be located anywhere in the program. A typical question of this type would be "How many WriteI statements are in this program?"
(2) All occurrences of a feature, which could only appear at specified points in the program. A typical question of this type would be, "Which procedure uses local variables that are not of type integer?"
(3) A specific feature, that could appear only once in the program, but could occur anywhere in the program. A typical question of this type would be "What expression is assigned to the variable Distance?"
(4) A specific feature, the identity of which could be determined by studying two separate sections of the program. A typical question of this type would be "What variables are declared in the main procedure but never used in it?" In order to answer this question, the subject had to study the data declarations of the main procedure and the body of the main procedure, separated by the other procedures and functions.

In this experiment, each subject would answer one question of each type for each program, each program appearing on a different screen configuration. The actual questions used are in the appendix; in each case, Question 1 is of type 1, Question 2 is of type 2, Question 3 is of type 3, and Question 4 is of type 4. Also, each subject answered the questions in the same order.

In summary, the independent variables were:

(1) Screen Configuration/Window Size (four levels)
   (a) 10-line configuration
   (b) 22-line configuration
   (c) 60-line configuration
   (d) 120-line configuration
(2) Question Type (four levels)
(3) Program (four levels)

We measured the following dependent variables:

(1) Correctness of answers
(2) Time to answer each individual question
(3) Total time to answer each set of four questions.
The subjects were 16 students in an advanced computer science course on human factors in computer and information systems. Each subject was given an instruction sheet which included a table of the Program Function (PF) keys used in the experiment. He/she was then told to concentrate on the specific questions being asked and not to try to understand the whole program. The terminal was set to the appropriate display, and the first question, written on a separate 3 x 5 card, was given to the subject. The subject read the question, and when she/he had understood it, pressed a PF key which caused the program to appear on the screen and the clock to begin timing. The subject would look through the program, scrolling if necessary, and followed the instructions to answer the question. Both the subjects' answers and the time involved were recorded on the computer. After each set of four questions had been answered, the subject recorded his/her subjective impressions about the configuration just used. Finally, after all four configurations had been used, the subject filled out a background survey.

The answers were graded such that questions of types 1 and 2 were worth two points, and questions of types 3 and 4 were worth three points. Partial credit was given for certain answers.

5. Results

When we examined the correctness scores (16 per display size) for all four programs, we found:

<table>
<thead>
<tr>
<th>Display Size</th>
<th>Mean Correctness Score</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6.00</td>
<td>1.76</td>
</tr>
<tr>
<td>22</td>
<td>6.94</td>
<td>2.46</td>
</tr>
<tr>
<td>60</td>
<td>7.44</td>
<td>2.20</td>
</tr>
<tr>
<td>120</td>
<td>7.75</td>
<td>2.19</td>
</tr>
</tbody>
</table>

Although the average correctness score is increasing, the difference is not statistically significant, as shown by a one-way ANOVA. Apparently all of the programs were not equally difficult. The transferable votes program produced statistically significantly lower scores than the other programs in all display sizes. Examining only the scores (12 per display size) on the programs of similar difficulty, we found:

<table>
<thead>
<tr>
<th>Display Size</th>
<th>Mean Correctness Score</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6.08</td>
<td>1.71</td>
</tr>
<tr>
<td>22</td>
<td>7.75</td>
<td>1.79</td>
</tr>
<tr>
<td>60</td>
<td>8.50</td>
<td>1.19</td>
</tr>
<tr>
<td>120</td>
<td>8.58</td>
<td>1.75</td>
</tr>
</tbody>
</table>

A one-way analysis of variance revealed statistically significant differences \((p < .01)\) for the factor of display size, and an HSD test revealed that both the 120-line and the 60-line displays were superior to the 10-line display. None of the other comparisons were significant by the HSD test.
The time results were analyzed by individual question type, and we received the following results (in seconds):

<table>
<thead>
<tr>
<th>Display Size</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>22</td>
<td>60</td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>120.4</td>
<td>170.9</td>
<td>130.6</td>
<td>107.6</td>
<td>(Mean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>44.4</td>
<td>122.3</td>
<td>71.7</td>
<td>48.4</td>
<td>(Std Dev)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>110.2</td>
<td>105.6</td>
<td>92.1</td>
<td>95.2</td>
<td>(Mean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>33.7</td>
<td>69.9</td>
<td>61.1</td>
<td>66.6</td>
<td>(Std Dev)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question Type</th>
<th>3</th>
<th>92.9</th>
<th>90.4</th>
<th>61.6</th>
<th>70.0</th>
<th>(Mean)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>60.4</td>
<td>78.7</td>
<td>25.1</td>
<td>49.1</td>
<td>(Std Dev)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>226.9</td>
<td>184.1</td>
<td>188.0</td>
<td>120.8</td>
<td>(Mean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>111.9</td>
<td>147.5</td>
<td>146.3</td>
<td>53.1</td>
<td>(Std Dev)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Time</th>
<th>553.1</th>
<th>551.0</th>
<th>472.3</th>
<th>393.6</th>
<th>(Mean)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>175.3</td>
<td>323.8</td>
<td>240.4</td>
<td>157.1</td>
<td>(Std Dev)</td>
<td></td>
</tr>
</tbody>
</table>

A two-way ANOVA found significant main effects for question type, \((p < .01)\) and display size \((p < .05)\). There were no significant interaction effects.

A count was also kept of the number of page turn commands each subject issued, according to display and question type:

<table>
<thead>
<tr>
<th>Display Size</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>22</td>
<td>60</td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>17.50</td>
<td>13.19</td>
<td>2.44</td>
<td>0.06</td>
<td>(Mean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.40</td>
<td>11.18</td>
<td>2.12</td>
<td>0.24</td>
<td>(Std Dev)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>21.38</td>
<td>10.38</td>
<td>1.44</td>
<td>0.25</td>
<td>(Mean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7.45</td>
<td>5.07</td>
<td>0.61</td>
<td>0.97</td>
<td>(Std Dev)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question Type</th>
<th>3</th>
<th>21.25</th>
<th>9.69</th>
<th>1.44</th>
<th>0.00</th>
<th>(Mean)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9.38</td>
<td>7.87</td>
<td>0.86</td>
<td>0.00</td>
<td>(Std Dev)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>53.81</td>
<td>38.50</td>
<td>6.88</td>
<td>0.00</td>
<td>(Mean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>33.70</td>
<td>29.90</td>
<td>9.95</td>
<td>0.00</td>
<td>(Std Dev)</td>
<td></td>
</tr>
</tbody>
</table>

The dramatic differences in the number of page turns is apparent. Most of the difference is due to the amount of scrolling needed to see the entire program. With a 120-line
display, no scrolling is necessary for these short programs. With a 10-line display, repeated scrolling is necessary to see the program even once. With type 4 questions, which require the study of two separate parts of the program, repeated scrolling is necessary with the 10-line display just to find the separate parts.

6. Subjective Reactions

Nineteen people filled out subjective evaluations of the screen configurations. These included the sixteen subjects plus three people whose data could not be used in the analysis due to external factors. These evaluations consisted of little more than free-response subjective answers to the question “What do you think of the display,” asked after each trial. There was a general preference for seeing as much code as possible. A sizeable minority, however, were not convinced that seeing 120 lines of code would be beneficial in all circumstances; they cited the dangers of clutter and of clumsiness. Nevertheless, almost all the subjects expressed a preference for configurations bigger than the traditional 24-line terminal. This was unexpected, because twelve of the respondents had reported using the traditional terminals almost exclusively and that it was their favorite size.

The observations of the experimenter tend to bear that out. When the subjects were using the twelve-line configuration, the subjects became generally frustrated with having to scroll through the program in so many small steps. This was especially noticeable when the subjects were answering type 4 questions, and they would have to go from one end of the program to the middle. In contrast, users of the 120-line display merely looked from location to location in a regular pattern, signifying that they were able to recall the location of specific points of the program on the screen. It should also be pointed out that the subjects were generally surprised to see the 120-line display fill up the entire terminal, particularly after they had seen two or three of the other displays.

7. Discussion

The correctness result supports the hypothesis that the bigger windows are better. We did not find a significant result when all four programs were included because the transferable votes program produced correctness scores that were significantly lower than those for the other three programs. When we focused on the three programs that produced similar scores, we found a statistically significant result.

The total time, with no question breakdowns, did not produce a significant difference. Again, we speculate that this is due to the range of times the various questions produce. The two-way ANOVA revealed a statistically significant difference among questions with type 4 questions taking significantly longer to answer than the others. Whether that is due to the larger screen, the absence of scrolling with these particular programs, or some other factor is an issue for further experiments.

The number of page-turns should decrease as the window size increases; subjects needed fewer page-turns to see an entire program if the window size was bigger. One might expect that with the 120-line display no subject would have to turn pages with
these programs, which have fewer than 120 lines. That is what happened; the five pageturns recorded are instances in which the subject hit the wrong key accidentally. It was speculated that the page turns would slightly decrease through question types 1-3 as the subject became more familiar with the program and then dramatically increase with question type 4, which was harder and required more page turns. This pattern appeared with the 22-line and 60-line displays, but did not appear with the 10-line display.

This may indicate that users may have trouble locating specific points in a program, as opposed to counting features which may occur anywhere, with a small display. Or, it may be due to the increased use of top and bottom commands with the ten-line display, which were counted in the page-turns as the maximum number of pages possible. The use of those commands also led to an occasional overshoot of the target and meant even more page turns. The top and bottom commands were most often used with the 10-line display.

A common pattern of looking at programs on the 10-line display was to slowly page through the program once to learn where everything was, and then take advantage of that knowledge to use the top and bottom commands later. A subject who moved from the end of the program to the second page using page-turns would need ten keystrokes, whereas turning to the top and then paging once forward required only two keystrokes.

8. Suggestions for Further Experiments

As mentioned earlier, the programs studied all had less than 120 lines, which meant that they could fit in the 120-line display without any page turning. The features of the editor were disabled, allowing the users only to turn from page to page. The font size was kept constant. The results of this experiment may not hold in different conditions.

The following are candidates for follow-up experiments:

1. Repeating this experiment with longer programs, so that even on the 120-line configuration, page turning is necessary.

2. Repeating this experiment, but enable the editor commands. Any "search" or "locate" command would be especially useful for finding tokens or expressions.

3. Repeating this experiment, but increasing the emphasis on program comprehension, rather than program reading. Replications with other tasks such as composition, debugging, and maintenance would also be worthwhile.

4. Allowing subjects to gain familiarity with larger screen sizes over a one week period before conducting the experiment.

In addition to the development of larger windows, there are other uses for the larger screens that should be studied. Some of them are given below.

Multiple windows. Instead of making one large window, use the screen to open up multiple windows. Each window can allow vision of different parts of one file or of different files. Issues to be looked at include that of synchronized scrolling vs. independent scrolling vs. fusion [Weldon, Norman, and Shneiderman, 1986] and that of tiled windows vs. overlapping windows [Bly and Rosenberg, 1986].
**Novel display strategies.** One of the uses of larger screens might be to provide additional information about a file without detracting from the ability to see the contents of the file. Such strategies include the hierarchical browser and embedded selection. [Shneiderman et al., 1986]

9. Conclusion

This study has demonstrated that even in a one hour period student programmers can become familiar with larger screen displays so that their performance in program reading tasks improves. Small displays, such as 10 lines, seem to inhibit performance because substantially increased page turning is required. These results support the current efforts to provide larger screen sizes with more information visible, but there may be a danger of overwhelming the user with too large a screen.

Acknowledgements: This work supported in part by IBM Federal Systems Division, Bethesda, MD. We appreciate the constructive comments of Kent Norman and Marvin Zelkowitz and the technical assistance of Carl Smith.

References


Weldon, Linda J; Norman, Kent L. and Shneiderman, Ben, "An Empirical Test of Display Strategies for a Multiple Screen Programming Environment" (1986)