U S A B I L I T Y R E P O R T

DATE: April 25, 2010

TO: Chris Hundhausen, CptS 443/543 Professor

FROM: ALVIS Team

RE: ALVIS Tools Usability Test

SUMMARY

Purpose and Scope

On April 20th through 22nd, we conducted a usability study of ALVIS Tools, a component of the ALVIS integrated development environment that enables novice programmers to explore algorithm design through a direct manipulation interface. In the study, participants explored a high-fidelity prototype and completed a set of tasks in order to get a feel for the software's functionality. Questionnaires and interviews were employed throughout the study sessions to elicit data that gave us insight into the extent to which ALVIS Tools supported what the participants wanted to accomplish.

Methods

We recruited four participants grouped into three sessions for this preliminary usability study. These participants were recruited from the Spring, 2004 CptS 121 course at WSU and paid 10 dollars for a one hour study session.

We ran the study in the WSU Visualization and End-User Programming Lab (VEUPL), where users interacted with the ALVIS Tools prototype running on a 1.79 gigahertz DELL computer with 1 gigabyte of RAM, a 64 megabyte ATI graphics card, and a 17 inch LCD monitor. User video and audio was recorded with a Minolta DiMAGE X20 2.0 mega pixel digital camera, and high resolution screenshots were captured with proprietary software.

We allotted 1 hour for each study session. At the beginning of the test, participants filled out a background questionnaire. See Appendix A for a copy of the questions and their responses. Next, they completed a warm-up exercise, in which they were given a brief description of the ALVIS Tools prototype software, and then asked to think aloud while they explored the software interface for 5 to 10 minutes. Appendix B presents an overview of the software.

After the warm-up exercise, participants were asked to complete a series of six task sets with the ALVIS Tools software. See Appendix C for a copy of the original task sheet that participants received for the test; Appendix D details the

Commented [cdh1]: Notice that throughout the report, references to the study materials are provided. MAKE SURE THAT YOU MAKE ALL OF YOUR MATERIALS AVAILABLE IN YOUR REPORT!

rationale for each task. As they worked through these tasks, participants were instructed to think aloud by verbalizing their thoughts and actions. If they became silent, or if it was unclear what they were up to, they were prompted to explain what they were doing.

Upon completion of the tasks, participants filled out an exit questionnaire that solicited their impressions of the ALVIS Tools software. Appendix E includes the exit questionnaire and participants' responses.

FINDINGS

Overview

Overall the ALVIS Tools design can be considered a success. The software received the following QUIS ratings (out of 9; See Appendix E for full details):

- Average Overall User Reactions : 6.67
- Average Screen: 7.36
- Average Terminology and System Information: 6.54
- Average Learning: 8.19
- Average System Capabilities: 6.40

Furthermore, where a usability requirement applied to a task (usability requirements were specified in the ALVIS Tools Design Document; a review is included in Appendix G. These requirements are applied in a time on task analysis reported in Appendix H), the design fulfilled that requirement 64.29% of the time.

We begin each subsection below with a brief description of the user subtask to which the problems listed in the subsection pertain. Next, in order of decreasing severity, we describe the associated usability problems, and provide "Severity" and "Scope" ratings[1] based on the empirical evidence. Following the problems, we cite evidence of, and a diagnosis of, each problem. Each subsection concludes with specific design recommendations that we believe will remedy the problems.

1. Lack of Functionality Visibility

Task:

> "Fill the rightmost array with variables." Two arrays are onscreen—the product of previous actions. The participant is tasked with filling an array specified as the result of a spatial relationship statement. The most efficient way to accomplish this task is to utilize the Auto-Fill tool.

[1] Severity indicates the level of difficulty that the problem caused users (1 = most severe). Scope indicates the range of users that the problem is likely to impact (1 = broadest). See Appendix E for precise definitions of these terms.

Commented [cdh2]: The "Overview" section presents an "executive summary" of your key results. This is where I expect you to present the **table** in which you assess each of your usability and user experience requirements vis-à-vis the data you collected. Remember: When this report was written, there was no such requirement. Nonetheless, Appendix H of this report presents such an analysis.

Commented [cdh3]: Look at how specific this analysis is. Can you do the same?

Commented [cdh4]: Notice how this report describes the task in which each problem occurred. Readers will find this to be helpful when they try to make sense of the problem.

Application of this tool to an array automatically fills all empty indices in the array.

Problems:

1. Tasked with filling an array with variables, users may choose to make variable instances with the Create Variable tool one by one and placing them into the array. They may not recognize the existence or purpose of the Auto-Fill Tool despite explicit tool tip information (Severity = 1, Scope = 2).

Evidence:

Two of the three participants exhibited this problem. Participant pair MB eventually discovered the tool with a little prodding; Participant R gave up on this task after filling only a single column.

Diagnosis:

It is possible that the Auto-Fill tool suffers from inadequate *visibility* (Norman, 1989). Another possibility is that the label "Auto-Fill" isn't suggestive of the tool's functionality.

Recommendations:

1. Redesign of Toolbox buttons to better afford their usage. The Auto-Fill Tool icon (that of a wand) did not appear to afford the action of filling. The new icon depicts a bucket; this may better afford 'filling' and is immediately recognizable to those who use graphics editing software regularly. Additionally including the common image of an array with the bucket may help to put the button icon and so the tool in context for the user.

Commented [cdh5]: Notice that each problem is stated in general terms ("users may…"). Notice also that each problem receives a severity and scope rating.

Commented [cdh6]: Most often, you state your evidence in terms of how many participants experienced the problem. In addition, you could provide here the counter location at which the problem occurred. At a minimum, a reader should be able to look up that information in your content log appendix.

Commented [cdh7]: Make sure to enlist design principles and theories covered in the course here. Note: I am including a sample diagnosis only in this section; however, your report should include a diagnosis of EVERY usability problem in your report.

Commented [cdh8]: Notice that each recommendation provides a concrete suggestion. This particular recommendation actually doesn't go far enough. Indeed, the authors should have provided an illustration of the new icon they designed. The more specific and concrete you can be, the better.

2. **Misdirecting Cue**

Task:

"Place a variable in the array cell at index (2, 2)." The participant is tasked with placing a previously created variable in a specific index of an existing array. To successfully complete the task, the participant needed to position the cursor's hotspot (focal point detected for software operations) within the appropriate cell and release the dragged variable picture.

Problems:

1. Users may not successfully place a variable in an array due to ambiguous relationship between cursor, dragged variable picture, and array cells (Severity = 1, Scope = 3).

Evidence:

One of three participants encountered this problem. Despite displaying both textual and graphical cursor feedback, Participant R searched a 3 column by four row array for 23 seconds before positioning the variable's picture over array cell (2,2). When he finally placed the variable, it snapped to an unintended array cell. The user did not recognize the hotspot on the cursor. This may be due to the large amount of visual information that accompanies the cursor in this type of situation: a cursor image, the variable icon, and textual output.

Commented [cdh9]: Notice that, in addition to stating the number of participants who experienced the problem, the author furnishes details of the episode in which each participant encountered the problem. This extra description makes the problem more tangible to the reader—a hallmark of an effective usability report.

Recommendations:

1. Rendering the variable in-place, where it would be positioned if dropped in an array, would greatly reduce the ambiguity in feedback regarding the insertion operation.
2. The combination of insertion cursor design and spatial relationship to the dragged picture makes the hotspot unclear to some users. The hotspot often ends up being neither in the middle nor a corner of the effective cursor (the cumulative presence of the cursor and dragged variable picture), as is most typical. A redesign of the cursor and orientation of the dragged picture may help solve this problem.
3. Some other method of determining array cell drop targets based on the variable picture position ad dimensions may help.

Commented [cdh10]: A better recommendation would have provided a picture of a redesigned cursor.

3. Lack of Functionality Visibility

Task:

"Modify the Create Variable Tool properties so that, when used, it will make variables with…" Participants are asked to modify a number of properties for the Create Variable tool. Users had to find the particular properties within a tabbed properties sheet and specify appropriate values.

Problems:

1. Tasks that require delving deeper than the first Quick tab in a property sheet suffer longer task execution times (Severity = 3, Scope = 2). (Compare Time on task item TT-41 and sub tasks in Appendix G with tasks that only access properties available on the Quick Tab)

Evidence:

Three of three participants encountered this problem. Completion times for those tasks requiring delving deeper into properties pages tend to be several seconds longer than those that require only properties available on the Quick tab.

Recommendations:

1. Design a cascading properties dialog system that behaves in a manner similar to context menus. A description of this design and a low fidelity prototype can be seen in appendix G of this report.

Commented [cdh11]: Now that's more like it: A whole appendix is dedicated to presenting a possible design change. I like it!

4. Inadequate Feedback

Task:

"Add that image to the Picture Gallery." Having created a picture with the Sketchpad, the participant is directed to add the picture to the Picture Gallery. To accomplish this, the user must add it from the "File→Add to Picture Gallery" menu option in the Sketchpad editor dialog.

Problems:

Users may not recognize the correspondence between the duplicated controls on the Quick tab and the other tabs in the property sheet (Severity = 2, Scope = 2).

Evidence:

After not locating the desired picture in the Picture Gallery, one participant drew his own, adding it to the Picture Gallery as requested. Upon completion, that participant opened the Picture Gallery yet again in order to verify that the picture was indeed added.

Recommendations:

1. The Sketchpad editor should automatically add an edited image to the gallery. If a user manually adds a picture through the dialogs menu or toolbar option, the result of the operation should be displayed in a message box.

5. Unrecognized Correspondence of Duplicate Functionality

Task:

> "Create a new variable with a picture of a red flower." The participant must first ascertain whether the picture exists or not. The picture intentionally does not exist, so the user must create a new picture. To do this, the participant needs to open the Sketchpad editor and design the flower picture.

Problems:

1. Users may not realize that there is only one picture gallery that can be accessed from multiple places. As a result, they may waste time searching that same picture gallery multiple times, each time believing that they have entered a different picture gallery (Severity = 2, Scope = 3).

Evidence:

> Two out of three participants encountered this problem. Tasked with creating a variable with a picture of a red flower, these participants
>
> 1. looked in Picture Gallery for a red flower.
> 2. gave up looking in the Gallery for the red flower picture.
> 3. looked in the Gallery again.
> 4. moved to the Presentation properties page and accessed the Gallery from there for a third time, expecting different results despite the fact that there is only one Picture Gallery. Multiple ways to access the Picture Gallery created the illusion of having access to more than one collection of pictures.

Recommendations:

1. The cascading dialog design (detailed in Appendix J) addresses this problem by keeping a single version of controls visible in a series of cascading dialogs.

6. Inadequate Feedback About Tool Mode

Task:: N/A

Problems:

1. The combination of cursor design and Toolbox button behavior was not sufficient to inform the participants of the current tool mode of the software (Severity = 3, Scope = 3).

Evidence:

Participants in two out of the three sessions encountered this problem. Participants MB accessed the Create Variable Tool properties by double clicking the tool button. In the end they did not realize that this also selects the tool. After modifying the tool properties, they activated the tool again despite the fact that it was already activated. (See CI-MB3, CI-MB4 in Appendix I for additional instances of this problem.)

Recommendations:

1. We have already implemented Toolbox buttons that indicate tool selection, larger color cursors. We plan to implement a Message box warning upon re-activating a tool.

APPENDIX A: BACKGROUND QUESTIONNAIRE RESULTS

C:\Documents and Settings\hundhaus\M

Commented [cdh12]: Notice how each supporting document is included as an appendix. It is OK to insert each item into the doc.

APPENDIX B: SOFTWARE OVERVIEW

C:\Documents and Settings\hundhaus\M

APPENDIX C: TASK SETS

Microsoft Word Document

APPENDIX D: TASK DETAILS

C:\Documents and Settings\hundhaus\M

APPENDIX E: QUIS RESULTS

Microsoft Excel Worksheet

APPENDIX F: USABILITY REQUIREMENTS REVIEW

C:\Documents and Settings\hundhaus\M

APPENDIX G: TIME ON TASK

C:\Documents and Settings\hundhaus\M

APPENDIX H: CRITICAL INCIDENTS

C:\Documents and Settings\hundhaus\M

APPENDIX I: USABILITY PROBLEMS

C:\Documents and Settings\hundhaus\M

APPENDIX J: CASCADING DIALOG LOW FIDELITY PROTOTYPE

C:\Documents and Settings\hundhaus\M