HCI Assignment 2

### Authors : s0927559 and s0943941

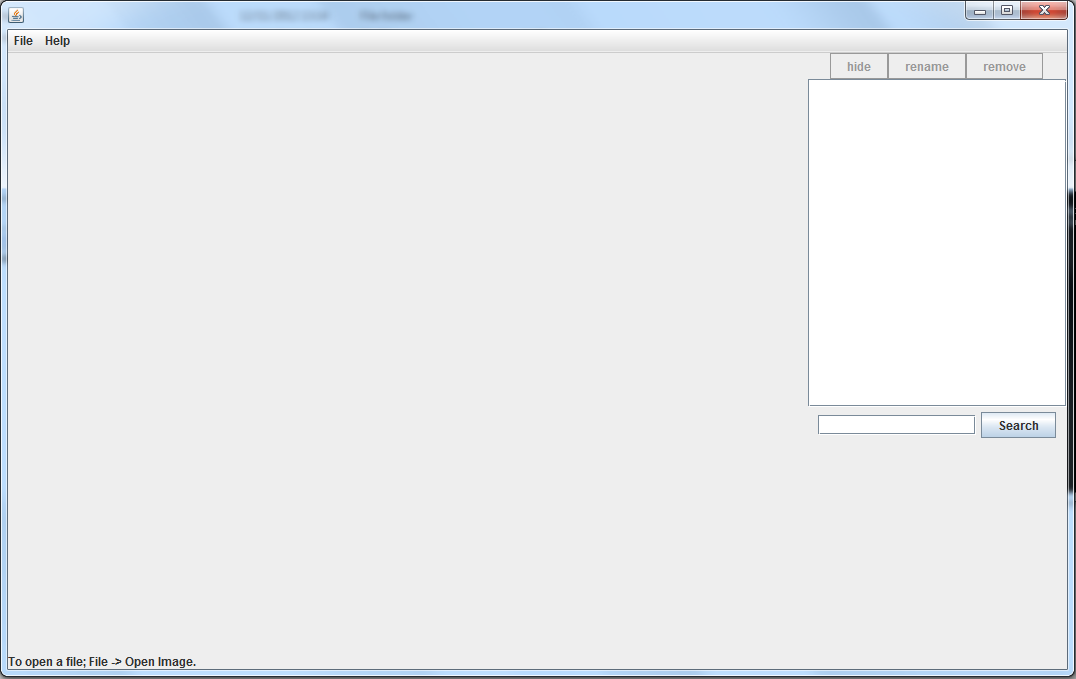
## Abstract

A formal analysis of the software ‘LabelIt’ with regard to HCI principles is presented below. First is a brief evaluation of the current design of the program, along with a critique of the application against design principles. Afterward, the results of a think-aloud evaluation are given, with a analysis of the results and the design intentions behind the task. A freeform user session follows this, with results from the questionnaire and conclusions from these tasks. As summary, the conclusion features a list of improvements suggested for the next iteration of the application.

The Current Design

This section presents the current design of the assigned application *LabelIt.* It discusses the strengths and weaknesses of *LabelIt,* and where these strengths and weaknesses meet or fall short of usability guidelines and design rules within its creation. Many of these strengths and weaknesses were further illustrated by the users in their test evaluation. This is discussed in the further sections of this report.

*LabelIt : Opening Screen*



The above image shows the opening state of *LabelIt.* The simple, single panel layout has a large area for the image to the left, and the control buttons for the labels on the right. It also has a small menu bar at the top, with buttons to deal with file actions (opening and saving images, and loading labels) and help actions (A help contents text popup, and a link to a youtube video which demonstrate the use of *LabelIt.* It also has a function to search for labels by their names with the label control area, and it has a prompt to open a file at the very bottom.

### Strengths

*LabelIt* has several strengths from its creation. It disables the use of buttons when they should not be used, making sure that the user does not get confused by buttons that do not need to be used in a given situation. The image loading from file only allows for the user to load images, and it attempts to check for changes and prompt for saving when needed. It also has a search function which will help a user with an image with a lot of labels on a single image.

Its design groups related actions to specific regions or into a menu list, and its simplistic design doesn’t distract the user from the goal of the program. These actions are also all very simple, and its obvious what each do. The help menu is also potentially useful to the user, and the help video clearly showcases how to use *LabelIt.* It also makes an attempt to cue the user to their first action with the prompt at the base of the image panel.

### Weaknesses

First, on a minor note, there are some features that are incomplete. Most buttons are disabled when they shouldn’t be clicked, but the search button is not. Also, the prompt at the bottom refers to “Open Image” while the actual button is labelled “Open File”. The button “About Us” in the help menu also does not work. While the “Open FIle” action is limited to images, the label loading action is not limited to XML files. The closing of the program does not work properly as well. The program window closes, but still runs in the background, keeping the java process active.

When the user opens an image, *LabelIt* always prompts to load a label file as well, which can be confusing to some users. You cannot load a label that doesn’t match the image, so it would have been better if the label and image were paired in some way, so that the user cannot make this mistake.

In terms of actual weaknesses or issues with *LabelIt*, a number of things were noted. When loading the images or labels, one issue is that the two actions are completely separate, requiring the user to remember: where the image is, and where the label is that is for that image.

When it comes to the simplistic design, although it doesn’t distract users from the goal of the program, users in our user testing did feel that several features were missing (This is further explained in the user evaluation section).

Another issue is that although the help video does show how to use *LabelIt*, it takes a lot of time, repeats functionality that is already demonstrated, it shows the help user doing things incorrectly, and shows *LabelIt* with an image already loaded, rather than starting with its initial state.

The functionality for the search function is incomplete. It only works on the exact match to the search term and only finds the first relevant option. The prompt text at the bottom isn’t obvious, and doesn’t go away after opening a file. Also, the colours chosen for labels appear to be arbitrary.

Once you get use to using *LabelIt*, you notice a few other issues. Placing points for labels requires the mouse to be still on click and release, rather than being able to move the mouse before a release, also there is no use of keyboard shortcuts (One is mentioned in the help contents, but it doesn’t work) . This means that there is a limit to how fast a user can work with *LabelIt.*

### Use of Design Rules in the Design

In this section we use the interface design rules described by Ben Shneiderman in *Designing the User Interface* to critique the interface of *LabelIt.*

#### Rule 1 : Strive for consistency.

This is partially met by *LabelIt.* Some of the actions are labeled and work in consistent ways, but some do not meet this. The action “Open File” and “Load XML Labels” in the file menu are related actions which aren’t consistent with each other in terms of labelling, and work slightly differently.

#### Rule 2 : Enable frequent users to use shortcuts.

*LabelIt* does not make use of any shortcuts, which caused some users minor irritation. It mentions a saving shortcut in the help contents, but this doesn’t seem to work.

#### Rule 3 : Offer informative feedback.

*LabelIt* offers some feedback to user actions. When a user wants to close a label, it does make the starting point different than the others, giving some users the idea that they need to click it again, but it is not explicitly said anywhere, except in the help. Also, the other minor feedback is that once a label is named, it is added to the list on the right side of the labelling panel.

The delete and renaming of labels show their effects immediately. Deleting the name and the label on the image, or changing the label name. Also, when a label is selected in the list, it’s label is highlighted. Also, a matching search highlights the correct label.

However, several other actions don’t give good feedback. The hide label button simply adds a “(h)” to the label, and the search does not make it obvious that you need to match the label name exactly. The save functions do not give any indication to whether the label or image was saved or not either. The naming dialog does not say what the cancel button will do, so the complete removal of the label can be confusing to some users.

#### Rule 4: Design dialog to yield closure.

Tied in with the informative feedback, some sequences of actions do not yield closure in an obvious way. The closing, naming, renaming and deleting label functions do give feedback to show the action is complete, but several others do not, or do so poorly.

The open image function always leads to *LabelIt* prompting to potentially open a labels xml file, when most of the time this isn’t needed. With some test users, this suggested to them that the image did not open correctly. The lack of feedback from the saving functions also do not give users closure.

#### Rule 5 : Offer simple error handling.

Beyond the simple core functionality of *LabelIt,* the program does not have many design features which prevent user errors. It does stop users from loading anything but images, but the loading labels functions does not restrict users to xml files. Also, it possible for users to use some of the functions in unexpected ways, because they aren’t prevented from doing so, or given feedback to say how the functions should be used.

#### Rule 6 : Permit easy reversal of actions.

*LabelIt* only meets this rule for one action: Allowing completed labels to be deleted, or labels being canceled and removed when they are closed and cancel is clicked in the naming dialog.

#### Rule 7: Support internal locus of control.

*LabelIt* does not give the strong sense that the user is in change and it is responding to it. While it does do this, there were several cases with the test users where they were wondering what they were required to do next, or how they should act, rather than see a response from the program.

#### Rule 8: Reduce short-term memory load.

Because of the nature of the program, there is not much that needs to be remembered by the user. *LabelIt* keeps this property as it is a simple, single window application. However, there is one place where the user must use memory. The dialogs for opening an image and opening a label are separated, which the mage and label are actually connected. So, the user must remember to save the label in the same location as the image, or remember where the saved a label for a specific image.

### 

### Use of Usability Heuristics in the Design

#### Heuristic 1 : Visibility of system status

Mentioned above in rule three of the design rules, “*Offer informative feedback”*

#### Heuristic 2 : Match between system and the real world

The design partially meets this heuristic, using terms such as hide, rename, remove and search for the label actions. In some places it also does meet this. The action to open an image is “Open File” instead of “Open image” and the opening labels action is “Load xml labels” instead of “Open Labels” to be consistent, or simply “Load Labels” to hide the fact from the user that the labels have a certain type.

#### Heuristic 3 : User control and freedom

The design partially supports undo functionality, but has no redo functionality. Dialogs have cancel functionality.

#### Heuristic 4 : Consistency and standards

As mentioned above, the text in the design are consistent with what users expect in most cases, but labels such as “Open file” instead of “Open image” and “Load xml Labels” instead of something else may make some users wonder.

#### Heuristic 5 : Error prevention

Discussed above in design rule 5: offer simple error handling.

#### Heuristic 6 : Recognition rather than recall

*LabelIt* meets this heuristic in most cases. Most actions are immediately visible to users, or are in one of two menus. However, the program needs to tie together the saved files of the image and the label, so that the user doesn’t have to remember their independent locations.

#### Heuristic 7: Flexibility and efficiency of use

*LabelIt* does not make use of any shortcuts which allow for an expert user to speed up their interaction. Unfortunately it also hinders users in some sense, because it forces users to pause, click and release to create a point, rather than being able to drag release.

#### Heuristic 8 : Aesthetic and minimalist design

*LabelIt* does have a minimalistic design, but some of the dialogs contain information that some test users did not expect, and caused them to act in error. The example is the open file (open image) dialog. After an image is chosen, rather than allowing to continue, or prompt for a label file in a non obtrusive way, it displays another popup saying that it can’t find the corresponding label.

#### Heuristic 9 : Help users recognize, diagnose, and recover from errors

*LabelIt* does not contain any error messages or any indication when there are problems.

#### Heuristic 10 : Help and documentation

*LabelIt* provides documentation which is easy for the user to reach. However, it has some misleading properties. It shows the program from its non initial state, and shows unnecessary steps. Also, for some users, it was difficult to use LabelIt without the help video, rather than being able to use it without the documentation.

User Evaluation

### Choosing Tasks

These tasks were chosen so that the tester would need to cover all the functionality that ‘LabelIt’ offers. For the evaluation, the key functionality of the program needed to be identified to structure the tasks around. The important functionality of Labelit was identified as the following:

1. Opening an image
2. Add a label
3. Name a label
4. Select a label
5. Loading labels
6. Saving the labels
7. View the help video
8. View the help contents
9. Use the search functionality
10. Rename the labels
11. Remove the labels
12. Cancel a label’s creation
13. Hide a label
14. About Us

To devise a list of tasks, we set the actions to model how a user would use such a program, while completing as many critical functions as possible. Key program functionality (opening an image, adding multiple labels and saving) was included twice within our list of tasks. This gave a fleshed-out understanding of how the user would approach the task. the relative complexity of the task, and to discover if the user would be able to recall the operations required for the task. One question was introduced twice (find a label) to determine which of the two functionalities the user would choose. Another question was repeated to ensure that the user understood the requirement to select the label needed to delete (once with the label pre-selected, the second time with the label not selected).

All other tasks captured the optional features of the program. The program was relatively simple in its design, with a limited amount of features, so complete coverage of the working features felt apt. Had the program been more complex, splitting the task into subsections to target a particular aspect would have been recommended, but with such a simple program, no such measures were necessary.

Certain tasks were excluded from our list. Anything involving longevity (i.e. save a label and load it a few hours later) was not included because our task involved first-time user interaction rather than multiple interaction. If the intended functionality was broken (i.e. ‘about us’ in the file menu did not work), we did not draw attention to it because we would learn nothing of importance. Finally, excessively repetitive tasks were dropped (i.e. did not include add forty labels and then search for a label) because this, while more revealing for certain functions (the user response to searching), would still provide a comparable response if simplified).

The task set was kept small to both keep the users engaged (a tired user will perform differently than a fresh one) and, had there been a larger sample size, to allow quick experimentation and analysis of the data.

The ultimate goal of this evaluation was to understand why the user performs the actions they do, what difficulties they in performing the task, and what their rationale was in approaching a task. The tasks were phrased to be direct commands, yet open enough to force the user to act instinctively/think about the problem (i.e. no information is given on how to solve the problem). The questions below are phrased with minor variations - the command ‘open the image’ was replaced with a specific file name. This allowed us to control the variable of

We consider the application to still be in the design phase, where any information obtained can be used to improve the software. With that in mind, the information received is quantitative in nature in addition to the verbal feedback received.

### Data

For the think-aloud evaluation, three users were selected. The data was collected using a single laptop and mouse configuration, running IOGraph software to capture mouse movement. The program was started for the user, and maximized to reduce interference with the rest of the computer.

Once the user was ready, the tasks were read out to them. The user was instructed to attempt to solve the problem on their own, but were allowed to answer questions. As the user completed the task, they were prompted to say what they thought they were doing. Only after they had completed the task were they allowed to proceed to the next task. Detailed notes were written during the evaluation about each task - a properly worded representation of these is produced below.

The list of tasks given is presented in the appendix.

|  |  |  |
| --- | --- | --- |
| **Task** | **Time Taken** | **Steps Taken** |
| **User 1** |  |  |
| Open image file | ~ 1 minute | Attempts to use search function to find file |
|  |  | Switches to file menu and opens image that way |
|  |  | Presses yes with uncertainty on open label prompt |
|  |  | Attempts to open label file by going to image and opening again. Assumes file invalid first time |
| Add label around an object | ~ 10 minutes | Clicking on screen doesn't work (bug when mouse moves after clicking - doesn't register) |
|  |  | Goes to load XML labels - "I don't know how to add a label" |
|  |  | Tries clicking multiple times. Clicks become faster as she becomes frustrated |
|  |  | Goes to help, starts watching video. Skips ahead in the video. |
|  |  | Tries load XML labels again |
|  |  | Goes to help contents |
|  |  | Attempts to click on blank rename entry |
|  |  | Reads help menu, proceeds to label around image |
| Name the label | ~ 5 seconds | Types immediately in lower case. Presses save. |
| Add another two labels | ~ 1 minute | Quicker this time, understanding what needs to be done. Occasional click-move bug. |
| Search for the first label | ~ 5 seconds | Immediately goes to search bar and performs required action. Matches on lower case |
| Select the second label you created | ~ 5 seconds | Clicks on correct label without hesitation. |
| Save the labels | ~ 20 seconds | Goes to file menu, pauses and expresses confusion about using save versus save as. Uses save as. |
| Delete the second label | ~ 5 seconds | Quick delete, no hesitation. Goes immediately to the correct button. |
| Load labels | ~ 2 minutes | Attempts to search for the deleted label |
|  |  | Attempts to load the deleted label name from the load XML labels option |
|  |  | Attempts to use control z to undo the function |
|  |  | Rereads help contents |
|  |  | Returns to load. Expresses lack of understanding. |
|  |  | Upon prompt, enters load XML Labels and reads each one by one. Finds correct label and loads. |
| Hide last label you created | ~ 5 seconds | Heads to hide button without waiting. |
| Discover what help functionality there is | N/A | Skipped, due to her already discovering it. |
| Open another image | ~ 1 minute | Finds image, presses yes upon prompt. Hits cancel and expresses confusion. |
| Add six different labels | ~ 4 minutes | Adds the labels, all with small mouse clicks. |
|  |  | Messes up drawing because of click-move bug, asks if there is an undo button. |
| Rename the second label | ~ 10 seconds | Finds and uses correct button. No hesitation. Clicks save. |
| Find the fourth label (given as a name) on the screen | ~ 5 seconds | Clicks on the list rather than use the search functionality. No hesitation. |
| Delete the third label | ~ 10 seconds | Quick response, took time to complain that she liked the label |
| Create label, but cancel before you finish naming it | ~ 30 seconds | Performs without hesitation |
| Watch help video | N/A | Skipped, due to her already discovering it. |
| Close the program | ~ 30 seconds | Attempts to go through file menu to exit |
|  |  | Realizes lack of exit there, closes with x |
| Total time taken | 22 minutes |  |
|  |  |  |
| **User 2** |  |  |
| Open image file | ~30 seconds | Brief confusion about opening file, unsure where to begin |
|  |  | Confused by label prompt - assumes that image cannot be found |
|  |  | Finds image again and tries to 'reload it'. |
| Add Label around the brain | ~ 9 minutes | Clicks around screen, but click-move bug prevents anything from happening. |
|  |  | Tries right clicking the screen. |
|  |  | Presses greyed out hide buttons. |
|  |  | Tries to load XML labels. |
|  |  | Single point appears on the screen, which she draws a line with. |
|  |  | Tries to finish the segment there, saying that she wants an arrow, rather than encircle the object. |
|  |  | Expresses frustration, and goes to help. |
|  |  | Complains that help isn't what she wanted, because it started at the bottom with how to save. |
|  |  | Attempts to click and drag. Mentions lack of understanding. |
|  |  | Finishes reading help, manages to finish off the first label |
| Name the label | ~ 5 seconds | Names it with a capital, and presses save with the mouse |
| Add another two labels | ~ 2 minutes | Labels others quicker. Still has difficulties due to move click. |
| Search for the first label created | ~30 seconds | Immediately uses search bar. |
|  |  | Spells with lower case, which does not match the uppercase label. |
| Select the second label | ~ 5 seconds | Immediately clicks on the label |
| Save the labels | ~5 seconds | Performs intuitively and immediately |
| Delete the second label | ~5 seconds | Goes to delete immediately, and removes. |
| Load Labels | ~1 minute | Goes to file -> Load XML Labels immediately |
|  |  | Complains about the amount of files in the directory |
| Hide Last Label You Created | ~ 5 seconds | Immediately finds and uses button to hide the label |
| Discover what help functionality there is | N/A | Already discovered help functionality, task skipped. |
| Open another image file | ~ 30 seconds | Opened file with ease. Still attempts to search for the image in the search bar. |
|  |  | Searches and finds image again for label prompt |
| Open another image file | ~ 3 minutes | Labels the image with shapes rather than interesting areas. |
|  |  | Names labels after shape, and complains about move-click bug (doesn't understand what is going on) |
| Rename the second label | ~ 5 seconds | Renames easily, no hesitation |
| Find the fourth label on the screen | ~ 30 seconds | Attempts to click on screen to select it. |
|  |  | Second attempt, clicks on the label list. |
| Delete the third label you created | ~ 5 seconds | Deletes easily without hesitation. |
| Create label but cancel before creation | ~ 5 seconds | Performs without hesitation |
| Watch help video | ~ 6 minutes | Finds without trouble. Watches majority of video, skipping past certain parts |
| Close the program | ~ 5 seconds | Closes the program with the 'x' |
| Total time: | 20 minutes |  |
|  |  |  |
| **User 3** |  |  |
| Open image file | ~ 2 minutes | Immediately goes to file -> open image |
|  |  | Attempts to type filename into File Name section and hit enter. Doesn't work. |
|  |  | Retries and selects file. |
|  |  | Pauses when the label prompt appears, goes back to reopen the image. |
| Add a label | ~ 6 minutes | Asks what a label is. |
|  |  | Presses on the screen several times, trying to get a result |
|  |  | First segment appears, but she doesn't know what to do next. |
|  |  | Starts circling an image soon after, but doesn't finish off the label. Assumes she's finished. |
|  |  | Tries naming it using the search bar. |
|  |  | Heads to help |
|  |  | Reads help, clicks button after reading prompt. |
| Name the label | ~ 5 seconds | Uses upper case, completes quickly. |
| Add another two labels | ~ 4 minutes | Returns to help |
|  |  | Attempts to add a single point and click on it to finish it. |
|  |  | Attempts to add another point and finish it. |
|  |  | Finishes with three points. Names with capital letters and spaces. |
|  |  | Finishes second label without trouble. |
| Search for the first label | ~ 30 seconds | Clicks on search bar, misspells search. |
|  |  | Corrects self, but doesn't match capitalization. Complains that it doesn't work. |
|  |  | Corrects self, matching capitalization. |
| Select the second label | ~ 10 seconds | Goes to help first, clicks on second one. |
| Save the labels | ~ 30 seconds | Checks help, attempts to use hotkey (A) |
|  |  | Presses save as, then attempts to check help while file loader is open. Doesn't work. |
|  |  | Saves succesfully |
| Delete the second label | ~ 5 seconds | Finds and deletes immediately |
| Load labels | ~ 1 minute | Attempts to open file instead of loading labels. This works despite the wrong intention. |
| Hide last label you created | ~ 30 seconds | Goes to hide button immedaitely, clicks and completes task quickly |
| Discover what help functionality there is | N/A | Skipped because was already discovered |
| Open another image file | ~ 30 seconds | Goes to open file quickly and loads without difficulty of before |
|  |  | Open label prompt, navigates to file again and reopens image. |
| Add six different labels | ~ 3 minutes | Completes with upper case and spaces, has issues with move-click bug. |
| Rename second label | ~ 5 seconds | Completes immediately, no issues |
| Find fourth label on the list | ~ 5 seconds | Clicks on the list (does not use search) |
| Delete the third label | ~ 5 seconds | Deletes the label without hesitation |
| Create label but cancel before creation | ~ 5 seconds | Creates label and presses cancel first time |
| Watch help video | ~ 6 minutes | Finds without difficulty, but complains about length |
| Close program | ~ 10 seconds | Exits with the X, saving before exit. |
| Total Time | 21 minutes |  |

### Evaluation of Data

#### First Use Issues

From the talk-out-loud testing, several key problems arose with multiple users. The three users initially chose to struggle with the program for several minutes before seeking help. The delay from when they started using the program to when they initially began labelling (task 2) was significant - on average eight minutes. Most of the issues did not come from the first task - opening a file is standard behavior to most computer users and this functionality does not differ much. The real difficulty lay with the labels and the drawing.

#### Labels

The inclusion of the labels caused the test users to have the most difficulty. Each user expressed confusion that the image did not immediately open, but instead showed a prompt. Because the name was the same as the image (but not the extension), the users assumed that the image had not opened. They navigated back to the image and tried reopening it. Upon opening the second image, two-thirds of the users repeated the erroneous behavior.

Labelling itself differed between users. Two of the three users espoused belief that the term ‘label’ did not bring to mind circling the image. Instead, they wanted to draw a single line pointing to a subset of the image with words near the stem. Such a belief requires a clarification of terms, or a visual metaphor to relay the idea that a label encloses the subject.

#### Adding Labels

As a task, adding labels took a large amount of the total task time as well. This was the result of the bug in Java where, if the mouse moves more than a few pixels from where it was clicked, it does not count as a click. This behaviour confused all test users, who repeatedly clicked on the screen to no effect.

Users reacted to the lack of observable feedback by attempting other possible solutions (typing entries into the search function, pressing empty buttons, attempting to reload). Had the program worked according to their intuition, they would have finished in minutes. Only after reading the help were they able to progress. Considering this bug affects the core aspects of the routine, and that each user encountered the bug, it is a critical flaw in HCI design.

On a different aspect, the preference of how the labels were drawn also varied. One user preferred dozens small clicks that traced the outline of the object perfectly. Another attempted to click and drag the cursor about the screen, akin to the use of a pencil. The third made large boxes that encompassed the majority of the image. However, the program only accounts for one style of drawing - large boxes that encompass the image. Users had different expectations about the program - a larger scale study would be needed to determine what the majority believes, and if attempts should be made to accommodate alternate concepts, or refine the program so that they alternative does not come to mind (via use of metaphors or renaming the ‘label’ concept).

The act of naming and cancelling labels, on the other hand, were tasks that users excelled at. The simple design and clear prompts gave users no difficulties.These features can be left untouched for the next iteration.

#### Saving

Only one user followed the help specifically through the tasks, though this lead to some confusion. During the ‘save label’ task, the help suggested that pressing the key A would save the image. No prompt was given for a successful save, so the user was left confused when the save did not work. Functionality should work, especially when given in the help documentation. More generally, the lack of feedback for saving caused confusion with two-thirds of users - such a reaction can be mitigated by explicit feedback that a ‘save’ action has occurred.

#### Loading

During the task of loading the labels, users approached the task differently. One user loaded the image again (which worked, raising the question of having separate buttons to perform the same functionality), and the other two attempted to find the labels. They complained about the amount of files in the workspace, which included document and image files. Reducing the amount of interference is key for usability - since we are loading labels, we should only be allowed to see .label files (at least initially).

#### Hiding, Renaming, Removing

The data showcases good performance time and an understanding of the three buttons located in the top right. For the tasks of renaming, hiding, and removing, users gravitated towards the area and used it successfully in a matter of seconds. In regard to the direct task, each user successfully clicked the label and then pressed the corresponding action button rather than the converse.

#### Searching

Two users also complained about the search bar requiring an exact match to find the label (matching capitalization and word length). This is an example of an industry standard that users expect to search bars to conform to. Fuzzy or subset searching is a feature in most search engines, and most searches do not require a match on punctuation to return something.

In response to the question designed to identify the prefered method of finding a label, two-thirds of users selected the label from the list. With such a small sample size, no evidence can conclusively be drawn from this information, nor was the test an adequate representation of all scenarios. A future test should involve having a large number of labels (~100) and repeating the test with a greater amount of users.

#### Help

Though the tasks were designed to direct the users to the help functionality, in each case the users discovered it on their own and used it (to varying degrees of success, given the faulty hotkey information contained within). The ease of access and near-immediate discovery by the users reveals highly valuable functionality and ease of use.

Two thirds of the users did make negative remarks about the length of the video. Nearly all of the users skipped about the video - either searching for critical information or attempting to find new information. Such behavior could be rectified by including a skip-to feature within the video so users could find out information about a particular subsection.

#### Exiting

The users exited from the program with little difficulty. One user, however, attempted to exit through the file menu (File -> Exit), as is standard with most applications. Deciding to include this functionality would ultimately require a larger study, but given the prevalence of the idea, it would be a useful inclusion.

#### Mouse Tracking

The movements of the user’s mouse were monitored during the task analysis. The reports are located in the appendix for viewing. For each of the three users, different patterns of use emerge. User 1 spent most time around the file and help systems; user 2 in the area below and to the right of the image, and user 3 in the center of the image. However, corroborating the information presented in the lecture, the majority of the mouse movements take place in the top left and centre parts of the screen.

This information suggests that the positioning of the image off to the left side might not be the most intuitive to user evaluation, but such information cannot be gleaned from mouse motion alone. Users reported preferring to keep their mouse away from the image when they were not labelling, and the second picture (given in the first part of the practical) featured a car in that position. Further information would be necessary to draw any major conclusions.

#### User Understanding

Users demonstrated clear ability to adapt and progress during the task list. With the repeated tasks, the time taken to perform the task decreased in all cases. Users also were able to apply previous useage to new scenarios. After the user deleted a label, they were able to hide a label with the same logic. None of the users had issues with functionality that was structured in the same sequence of events.

Quantitative Analysis

To complement the understanding of how the users behaved, and to see what functionality would be used without prompt, the system was given to five different users with the instructions to freely interact with it. Once the user had decided they had finished, they received a questionnaire designed to gauge their feedback on the system.

The resulting data would be used to understand the user’s impressions of the software (its appearance and its functionality), where the program excelled and lacked from how the user behaved without restraint, and to see incongruities between user performance and evaluation.

The questionnaire was designed as a qualitative rating system, with a series of closed questions pertaining to the important aspects of system design - difficulty of learning curve, ease of use, visual appearance, errors (or perceived errors), help, and overall impression of the software. We structured the questionnaire to be completed in similar sections, with each major component receiving its own rating (between 0-4 or 0-3, depending on the style of question).

Lower values were chosen to represent negative views, so that the greater the total value, the better the system has performed. The total score is out of 147. Had the sample size been larger, or the testing focussed on a particular area, the total amount of questions would have been reduced.

The questionnaire in full is reproduced in the appendix below.

The features are the same set considered for the think-aloud evaluation.

* + 1. Opening an image
    2. Add a label
    3. Name a label
    4. Select a label
    5. Loading labels
    6. Saving the labels
    7. View the help video
    8. View the help contents
    9. Use the search functionality
    10. Rename the labels
    11. Remove the labels
    12. Cancel a label’s creation
    13. Hide a label
    14. About Us

We considered having save and save as as two separate functions, but the overall functionality of the two actions are similar enough, and ubiquitous in everyday applications. If the user uses one, we consider them able to access the other. ‘About Us’, though not functional, is considered here as a feature and was counted if the user encountered it. This was for completeness rather than functionality.

The same setup used for the think-aloud evaluation was used for . The key difference was that the users were given no prompts and no feedback, and allowed to finish whenever they wanted. No mouse tracking took place for this section. The data set comprises of five individuals and all of university education and age. Note that the sample is not of a valid size, nor unbiased in range of individuals chosen. Also note that the testing and questionnaire were given without a pilot study beforehand - any shortcomings were not ironed out through iterative design. Despite these limitations, valuable information was received. The results are displayed below:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 0 | 1 | 2 | 3 | 4 | User 5 | User 4 | User 3 | User 2 | User 1 |
| **First Time** | Open Image |  |  |  | 3 | 2 | 3 | 4 | 4 | 3 | 3 |
|  | Add Single Label |  | 1 | 1 | 3 |  | 3 | 3 | 3 | 1 | 2 |
|  | Name Label |  |  |  |  | 5 | 4 | 4 | 4 | 4 | 4 |
|  | Save Labels |  | 1 |  | 1 | 3 | 4 | 4 | 1 | 4 | 3 |
|  | Find Label |  |  |  | 1 | 4 | 4 | 3 | 4 | 4 | 4 |
|  | Remove Label |  |  |  | 1 | 4 | 4 | 3 | 4 | 4 | 4 |
|  | Rename Label |  |  |  | 1 | 4 | 4 | 3 | 4 | 4 | 4 |
|  | Hide Label |  |  |  | 1 | 4 | 4 | 3 | 4 | 4 | 4 |
| **After Some Time** | Add Single Labels |  |  |  | 2 | 3 | 4 | 3 | 4 | 3 | 4 |
|  | Add Multiple Labels |  |  |  | 2 | 3 | 4 | 4 | 3 | 3 | 4 |
|  | Name Label |  |  |  |  | 5 | 4 | 4 | 4 | 4 | 4 |
|  | Load Image |  |  |  | 4 | 1 | 3 | 3 | 4 | 3 | 3 |
|  | Load Labels | 3 |  | 1 | 1 |  | 0 | 3 | 2 | 0 | 0 |
|  | Save Labels |  |  | 1 | 1 | 3 | 4 | 2 | 3 | 4 | 4 |
|  | Find Label |  |  |  | 1 | 4 | 4 | 3 | 4 | 4 | 4 |
|  | Remove Label |  |  |  |  | 5 | 4 | 4 | 4 | 4 | 4 |
|  | Hide Label |  |  |  |  | 5 | 4 | 4 | 4 | 4 | 4 |
|  | Rename Label |  |  |  |  | 5 | 4 | 4 | 4 | 4 | 4 |
| **Errors** | Add Label |  | 1 | 1 | 3 |  | 1 | 2 | 3 | 3 | 3 |
|  | Find Label |  |  | 1 | 4 |  | 3 | 2 | 3 | 3 | 3 |
|  | Open Image |  |  | 3 | 2 |  | 2 | 3 | 2 | 2 | 3 |
|  | Save Label |  |  | 1 | 4 |  | 3 | 2 | 3 | 3 | 3 |
|  | Load Label |  |  | 1 | 4 |  | 3 | 2 | 3 | 3 | 3 |
|  | Hide Label |  |  | 1 | 4 |  | 3 | 2 | 3 | 3 | 3 |
|  | Remove Label |  |  |  | 5 |  | 3 | 3 | 3 | 3 | 3 |
|  | Rename Label |  |  |  | 5 |  | 3 | 3 | 3 | 3 | 3 |
| **Error Overall** | Severity |  |  | 1 | 2 | 2 | 3 | 2 | 4 | 3 | 4 |
| **Appearance** | Image Display |  | 1 | 1 | 3 |  | 1 | 2 | 3 | 3 | 3 |
|  | Label List |  | 1 | 1 | 3 |  | 1 | 2 | 3 | 3 | 3 |
|  | File Menu |  |  | 2 | 3 |  | 2 | 2 | 3 | 3 | 3 |
|  | Popups |  | 1 | 1 | 3 |  | 1 | 1 | 3 | 3 | 3 |
|  | File Loader |  | 1 | 1 | 3 |  | 2 | 1 | 3 | 3 | 3 |
|  | Label While Drawing | 1 | 1 | 1 | 2 |  | 1 | 0 | 2 | 3 | 3 |
|  | Label After Drawing |  | 2 | 1 | 2 |  | 1 | 1 | 2 | 3 | 3 |
|  | Label When Highlighted |  |  | 2 | 3 |  | 2 | 2 | 3 | 3 | 3 |
| **Help** | Video | 2 | 1 |  | 2 |  | 0 | 1 | 0 | 3 | 3 |
|  | Text | 1 |  | 2 | 2 |  | 2 | 0 | 2 | 3 | 3 |
| **Overall Impressions** | Functionality | 1 | 1 | 1 | 2 |  | 1 | 0 | 2 | 3 | 3 |
|  | Feature Amount |  | 2 | 2 | 1 |  | 1 | 3 | 1 | 2 | 2 |
|  | Usefulness |  | 2 | 2 | 1 |  | 2 | 1 | 1 | 2 | 3 |
|  | Ease of Use |  | 2 |  | 2 | 1 | 2 | 4 | 2 | 3 | 3 |
|  | Expectations |  | 1 | 3 |  | 1 | 1 | 2 | 2 | 2 | 4 |
| Total Values |  |  |  |  |  |  | 109 | 104 | 123 | 129 | 136 |

#### Summary of Users:

Average Time Spent with System: 12.8 minutes

Average Features Found: 9/14 (rounded down)

Features Missed Most: Load Label (3/5 users), Cancel (4/5 users), About Us (3/5 users), Search (3/5 users)

Average Images Opened: 2 (rounded down)

### Evaluation of Data

#### Average Data

From the data, several conclusions can be drawn. Some of these are drawn solely from the selective sample size, and should not be considered applicable on a larger scale.

The users avoided certain functionalities more than others. These can be explained as the result of the short length of the study. In the short time, there isn’t as much incentive to return to a previous image, or to learn about the program. The lack of searching is, as in the think-aloud evaluation, a function of the small size of labels. Finally, about us does not pertain to the program use, so we can discard its importance. Overall, the users discovered the crucial functionality of the program and used it as intended.

### Survey

#### Ease of Use:

In regard to ease of first use, the responses were similar across all users. The task of adding a label for the first time was rated as being more difficult than any other of the initial tasks. This is likely due to the click-move bug mentioned above. All five users, during observation, experienced the bug, preventing them from drawing on the screen initially. While some users adapted faster than others, users 1 and 2 in particular struggled to use the application and were forced to refer to the help. The prevalence of this bug should be considered critical for fixing in the next iteration.

Ease of use after a period of time improved upon the scores given on the initial use scores. The only scores that were low in this section was loading of the labels, which was not present in the initial section. The lowness of this score can be attributed to misunderstanding of what a label file is. Of all the freeform users, only user 4 understood the importance by trying to load a different file and receiving an error.

The increase of scores demonstrates that the users believed they were learning. This is backed up by data - users spent, on average, 7 minutes on the first image and 5 minutes on the second. Some users even used 3 images. This means that they reapplied the functionality of the first image to the second in less time, and felt less need to spend a great deal of time label the image.

During the opening of the file, there was some confusion after the prompt to find the label file. Initially, each user pressed yes to find the label, and either opened the image again (users 1,2,3 and 5) or attempted to open a different document (user 4). Restricting the file loader to only show label files would solve both of these problems, and simplify the interface.

Finally, each of the users highly ranked the ease of ‘selecting, searching, hiding, renaming and removing’ in both initial and long term applications. This corroborates the data from the think-aloud section that the users found these tasks to be well-designed and simple.

#### Appearance

Concerning the GUI impressions, the majority of users found the application ‘pleasing’. A flaw with close-ended questionnaire is that the opportunity for an explanation is not given. The two users who rated the GUI less than ‘pleasing’ did not provide the information for why they held that opinion. Subjective tastes like GUI should be approached from both a rating system and a (potentially optional) feedback system.

#### Errors

During user 1’s testing, an instance with regard to saving occurred. The user labelled her image and switched to a new image. The changes she had made to the image were unsaved from an earlier iteration. She decided, a few minutes later, to return to the initial image. She attempted to reload the labels when they did not appear, but they were gone. In frustration, she tried again, with the labels appearing again on the screen. She stopped her evaluation soon afterwards.

However, when user 1 took the questionnaire, she ranked that she had encountered no errors. This highlights the potential flaw of these surveys - their subjectivity. The user did not realize that she had encountered an error, or forgot. Such surveys must be weighted in regard to their value - if a user ranks everything too highly, there is a chance that they are subject to favoritism or bias. Such a cognitive bias can be seen in the values for ‘add label’ - despite the click-move bug, hardly anybody recognized it as a bug.

#### Help

The valuation of help differed greatly between the users. Certain users, who were struggling initially (users 1 and 2), valued the help much higher than the individuals who were able to figure it out on their own. The spread of information makes this less important, but since the help does not actively interfere with the users interaction with the program, it should remain as is.

#### Overall

The application scored a total of 600/735 - a respectable score. The values can be improved by implementing the changes mentioned above - correcting the two important bugs, and refining the approach to labels. If the appearance needs to be improved, another survey with open-ended questions should be administered.

Conclusion

Various mentions and suggestions have been made throughout this report as to methods of improving the application. A list of important changes is recommended here, as the result of the formal analysis, the data gathered from the think-aloud evaluation and the questionnaire.

Fix the bug where, if the user clicks and moves the mouse before releasing, the line does not draw. Functionality critical to the application should be implemented and fully tested. Because this affects every user on a high occurrence, it should be replaced. Also, enable the use of shortcuts for actions. This is the direct application of the HCI design principles. Both of these will allow users to use *LabelIt* more efficiently, and reduce the number of interactions required.

Eliminate the separation of label and image, and delete the prompt. Every single user had initial difficulty with the prompt to find the label, partially due to assumed behaviour (errors will appear in a similar if the file cannot open in most applications) and such default behaviour should be assumed (it appears on every new opening). The case where the user wants to add labels occurs far less than the case where the user opens a new image. This can help eliminate errors, and reduce the memorisation that the user needs to perform.

Shift the image toward the centre of the image. Most of the user mouse activity is concentrated towards the center of the screen.

References

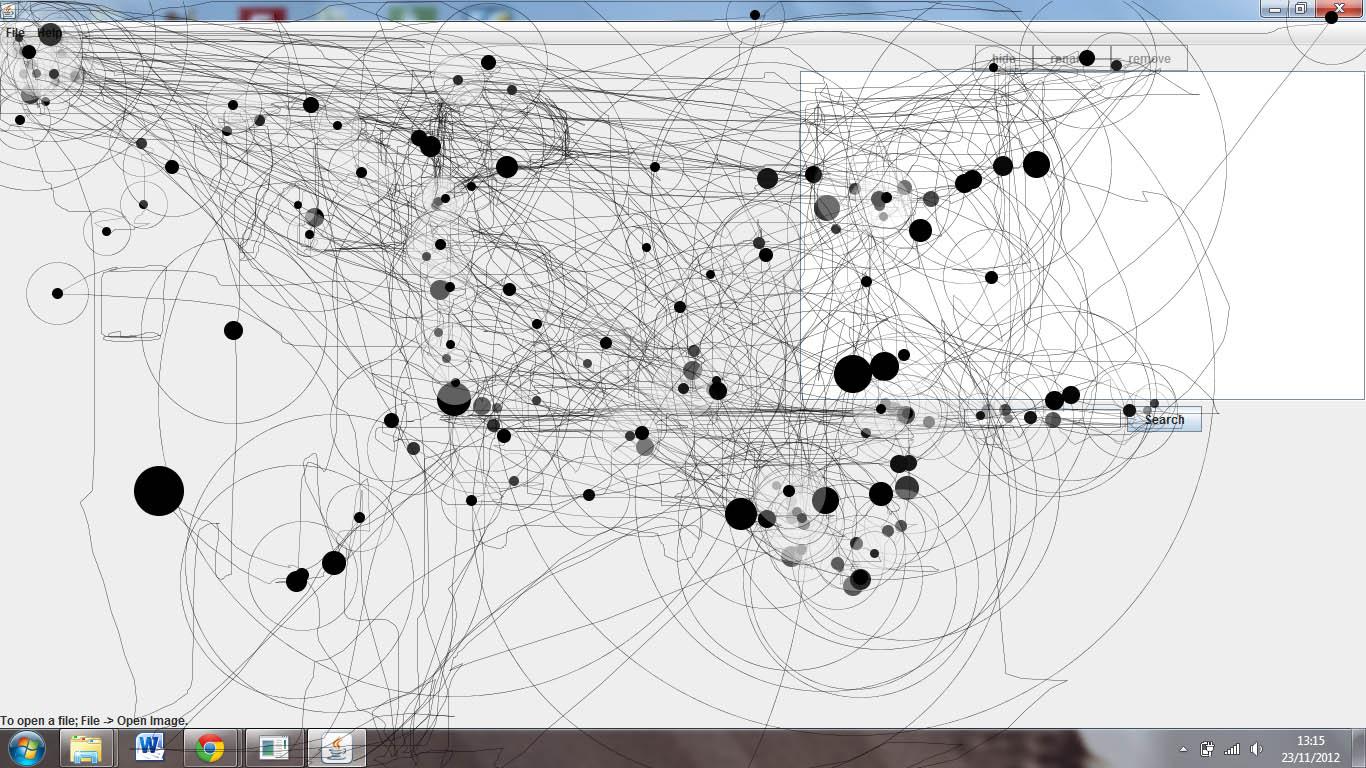
IOGraphica - <http://iographica.com/>

<http://www.useit.com/papers/heuristic/heuristic_list.html>

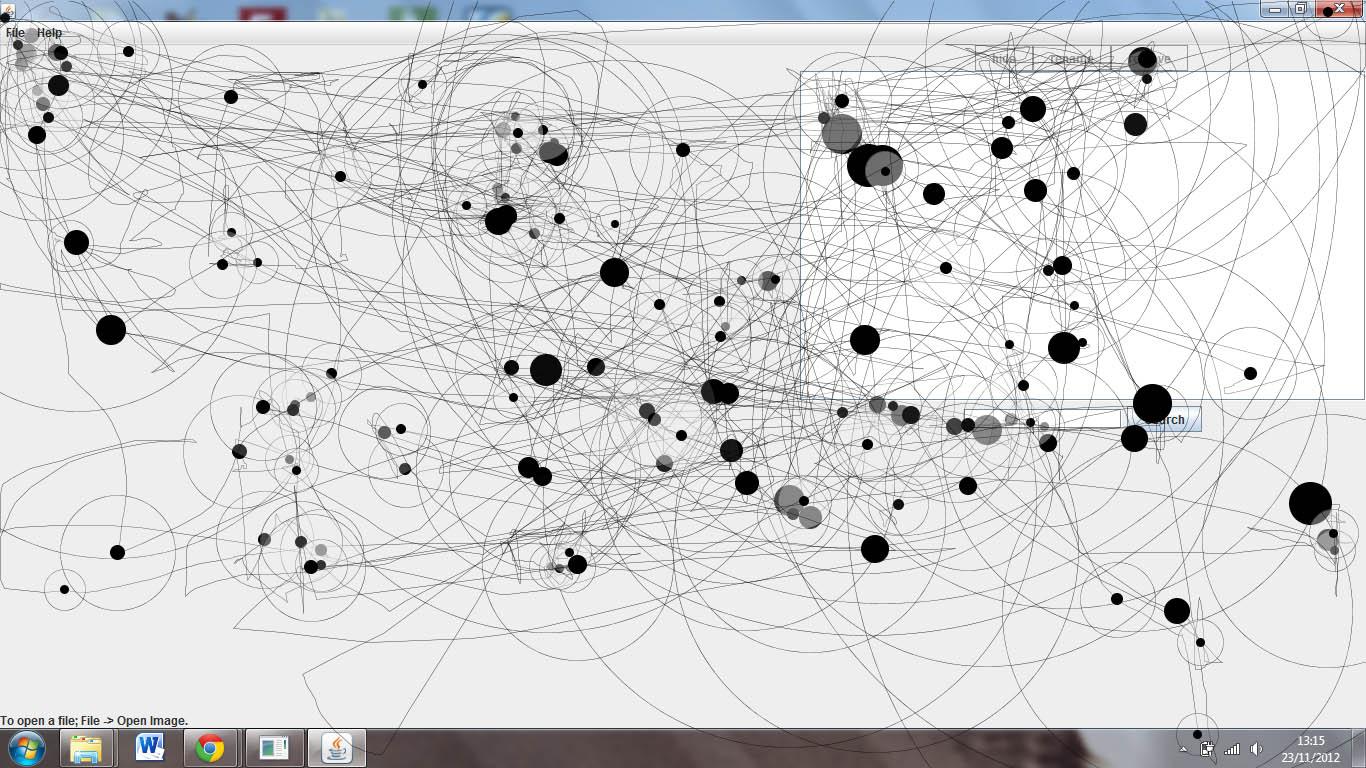
<http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html>

Appendix

User 1 Mouse Tracking



User 2 Mouse Tracking



User 3 Mouse Tracking

