ABSTRACT
In this paper, we describe the design, implementation, and evaluation of Disorganizer, an application for managing personal notes, ideas and reminders. Disorganizer attempts to bring the advantages of a paper-based organizational system into a computer-based system. It utilizes a novel interface based on zooming. We found that the approach generally worked well, although much future testing is needed.

Author Keywords
note taking, user evaluation, zooming.

ACM Classification Keywords
H5.2 User Interfaces: Evaluation/methodology, Interaction styles.

INTRODUCTION
A common task is managing personal notes, such as ideas and reminders. Many people prefer to use ad-hoc paper-based systems for this task, such as sticky notes, or simply papers arranged on a desktop or other work surface [2]. These systems take advantage of the spatial arrangement of the papers as an organizational mechanism.

Paper-based systems are cheap, widely available, easy to use, and perform well enough, most of the time. However, there are some limitations that a computer-based system could address. Physical papers degrade with time and use, and may eventually become unreadable. The arrangement of the papers can be unintentionally disturbed, thus destroying their organization. A computer-based system can provide functionality for quickly searching, copying, sorting, saving and sharing notes.

A significant advantage of paper is that it is unstructured. Most computer-based systems impose some type of categorical organization. This is a difficulty to users for two reasons. First, some notes will be difficult to categorize. Second, the effort required to categorize a note, even if relatively small, may be too great, particularly if the note is relatively unimportant, or likely to be short-lived. Previous studies indicate that the effort required to create a note is a major problem with computer-based systems [2,3].

We attempted to design Disorganizer as a computer-based system that retains as many of the advantages of a paper-based system as possible.

DESIGN
The design of Disorganizer began with user and task analysis. This was followed by an iterative design process. Disorganizer was evaluated by paper prototyping [6], heuristic evaluation [4,5] of a computer-based prototype, and user evaluation of a working version of the program. The design was revised after and during each evaluation.

We began with the assumption that we would design Disorganizer based on the metaphor of paper notes, and thus that the interface would present objects whose behavior was similar to physical paper notes. We made this decision because we believed it would make it easier to meet our design goal of retaining the advantages of paper notes.

To address some of the disadvantages of paper notes, we explored an interface based on zooming. We hoped to provide an interface that would scale up to an arbitrarily large number of notes while avoiding excessive overlapping of notes and retaining the notes original spatial relationships.

Target Users
We designed Disorganizer for adult users, of either gender. We assumed that users are comfortable with computers in general. Although such an application would more likely be used by someone who uses his or her computer every day, we also tried to make Disorganizer easy enough for a novice to use, after a short learning period.

More specifically, we targeted users who keep a large number of notes, but prefer not to use current systems that impose a strict organization. The typical user of Disorganizer will be someone with a “messy” desk, layered with various papers and sticky notes of all sizes and colors.
Task Analysis
We identified several critical tasks, based on an analysis of paper-based note systems.

Add an Item.
This task is very common, and must be relatively quick. The user must be able to easily decide the placement of a new item.

Notice Important Items.
The user must be able to easily notice notes that must be acted on in the near future. Ideally, this should happen without having to explicitly search for urgent notes.

Find an Item by Context.
The user must be able to find an item based on its context, such as the placement of the note, and what other notes are nearby.

Find an Item by Content.
The user must be able to find an item based on the content of that item. User interviews indicated that, when using paper-based systems, users often search both by content and by context, sometimes simultaneously.

Straightening Up.
The goal of this task is to modify the organization of a system to make other tasks more efficient. In user interviews, this task was uncommon, and badly supported by paper-based systems.

Remove Items.
The user should be able to remove unwanted notes with minimum disruption to the organization of the remaining notes.

Design
The main area of the Disorganizer is the desktop (Figure 1). The desktop contains all the user-created items. These include notes, and groups, which may contain notes as well as other groups. The desktop has an appearance similar to a paper-based notes system, where notes are placed on a physical desktop or bulletin board.

Notes.
The desktop contains several different types of notes, such as bills, ideas, and reminders. The shape and color of a note indicates its type. We confirmed, through paper prototyping, heuristic evaluation and user testing, that the primary means by which testers distinguished note types was shape and the secondary means was color.

Each note contains arbitrary text, which is fully under the user’s control. We considered including specific fields in the notes (for example, an “amount” field in bills). Some comments during heuristic evaluation and paper prototyping also suggested this change. We felt that this would complicate the task of adding a new note, which, according to our task analysis, should be as simple and easy as possible. However, some note types do contain an optional due date field, which may be used for sorting.

Notes are created via the “New Note” menu, which displays the shape and color of the note types (Figure 2) along with its description. Originally, each note type had an associated icon (such as a finger with a bow tied around it for a reminder), but heuristic evaluation revealed that this was confusing and added no value.
Figure 3. A group sorted by due date.

When notes are created, they appear in the center of the desktop. Each new note is offset slightly from the last, so that the previous note will not be hidden completely. This offset was added in response to heuristic evaluation.

Groups.
The user may create groups on the desktop. These groups may contain notes, and even other groups. This feature allows organization that a paper-based system does not.

However, unlike many computer-based systems, such as the familiar file browser interface, the contents of a group are always visible.

The contents of a group can be placed freely, or can be sorted according to due date (Figure 3). The group itself can be moved by clicking and dragging on the “move” icon in the upper left, and resized via the icon in the lower right.

Users may either first select items for grouping or create empty groups and then drag items into the new group. During paper prototyping, empty groups were not an option. If a user tried to create a group without first making a selection, they got an error message. However, even after rewording this message multiple times, the users were still confused by it. This directly influenced our decision to allow empty groups.

Zooming.
Notes and groups may be zoomed. A zoomed item expands to take up a large part of the desktop (80% in the case of groups), while all other items are squished and moved off to the side (Figure 4). Notes may be zoomed in order to edit or view them, while groups simply become larger, so that their contents may be more easily manipulated and viewed.

We chose this type of zooming rather than a simple magnification because it allows all items to remain visible and moveable.

Zooming does not affect the relative spatial arrangement of items. If the user zooms an item, and then zooms back out, all items will be in the exact position they were before the zooming.

Figure 4. The desktop zoomed in on a note.

The user may zoom any item by double-clicking it. In addition, groups that are not currently zoomed have a zoom-in icon in the upper-right corner, which may be single-clicked. Items that are zoomed (both notes and groups) have a zoom-out icon, or alternatively, may be double-clicked again to zoom out. In the original design, notes did not have a zoom out icon. This was changed following heuristic evaluation.

Heuristic evaluation and user testing revealed that users preferred to directly double-click on the item they wished to zoom, regardless of whether another item was currently zoomed. This was not possible in the earlier prototypes, and was added in response to this feedback.

Movement.
Notes can be moved by clicking and dragging. They may be freely moved in or out of groups. Both notes and groups can be moved regardless of whether an item is currently zoomed. If a group is zoomed, items may still be dragged in or out of it. If this feature were not available, the interface would be highly modal, and would require the user to repeatedly zoom out and zoom back in to accomplish many tasks.

Scaling.
Notes and groups are automatically scaled according to how “crowded” the desktop, or their containing group, becomes. If a group contains many notes, the notes will be shown smaller. This was necessary to avoid excessive overlapping of notes while still maintaining the spatial relationships between them.

Animation.
All movement (including zooming, resizing, scaling, deleting) is smoothly animated. This provides feedback,
which is particularly useful for novice users, who could otherwise be confused by the large changes in position that happen during zooming. We decided that this additional clarity outweighed the inefficiency of forcing the user to wait for the animation each time he or she zooms an item.

**Searching.**
The user may search the contents of notes by entering text in the search field on the toolbar, and clicking the adjacent “Search” button. In response, all notes that match the search are highlighted in red (Figure 5).

The red highlight distinguishes from selected notes, which are highlighted in yellow. Some comments from heuristic evaluation indicated that this two-color scheme may be confusing. However, we decided to keep it because it proved useful during evaluation. For example, a user could run a search, and then select one note from the search results to print out. The selected note would then show both red and yellow highlights, making it clear which note would be printed.

The “Clear Search” button on the toolbar clears the search field, and also removes the highlighting from the previous search results. In the earlier prototypes, this button did not clear the search field, but the heuristic evaluation found this to be confusing. In addition, if the user performs a new search, the results of the previous search are first cleared. This feature was also added after heuristic evaluation.

**Trash and Undo.**
Many of the evaluators, during heuristic evaluation, asked for an undo feature, or a way to retrieve deleted items. Following these suggestions, we added both a comprehensive undo system, and a “trash” from which deleted items could be both viewed and restored (Figure 6).

We chose to implement both of these features because they potentially serve different needs. Undo is primarily useful for reversing recent actions. Trash is useful for retrieving deleted items without reversing any actions that may have occurred since the item was deleted.

**IMPLEMENTATION**
Disorganizer is implemented in Java, and is based on the Piccolo framework [1]. Piccolo is a framework for building “Zoomable User Interfaces”. Piccolo programs build a hierarchy of “nodes”, each of which is capable of drawing itself at arbitrary scale. Piccolo provides support for smoothly animating nodes between different transformations, thus providing the appearance of smooth zooming.

Each Disorganizer item (note and group) is implemented as a Piccolo node, with several subsidiary nodes responsible for drawing things like move and zoom icons.

The choice of framework proved to be an important design decision. The advantages of Piccolo are that it made it possible to implement many of the zooming features which would have been far too difficult to implement entirely by ourselves.

However, it is difficult to integrate Piccolo objects with the standard Java GUI components. In many places, behaviors that are handled automatically by the standard components must be programmed “by hand”. This limited our flexibility, as simple features like adding a text box to a node would become a major undertaking. The standard components handle a great deal of behavior (such as selection) that is difficult to get right with the Piccolo objects.
EVALUATION

We evaluated Disorganizer with three test users, who were recruited from the workplaces of team members.

Subjects.
All subjects were administered questionnaires prior to testing. These questionnaires assessed how often the subject uses a computer, and what kind of organizational systems (both paper and electronic) he or she uses for personal notes.

All three subjects were male. Two subjects were in their 20’s, while one was 50. The two younger subjects were both students, while the older was an office manager. The two younger subjects most often used a laptop, while the older used a desktop PC. All three subjects stated that they use a computer almost every day.

Two of the three subjects stated that they tend to print out most of the notes they receive electronically. The third stated that he prints out ideas and reminders, but not other types of notes. The three subjects used varying means of paper-based organization, including bulletin boards, notebooks, and files.

The subjects stated that they receive or create personal notes (paper or electronic) at rates ranging from once a week to nearly every day.

Briefing.
Subjects were given a short printed briefing, which described Disorganizer and the general purpose of the application, but gave no specific instructions on how to accomplish any task within it. In particular, we did not explain how to zoom or move an item, or even that it was possible to zoom.

Subjects were provided a Disorganizer desktop that was already populated with notes and groups. This desktop is shown in Figure 1. This was done so that we could evaluate the performance of users on a reasonably complicated desktop, without forcing them to first repetitively add notes and groups. This rationale was explained to users, if they asked.

Tasks.
Each subject performed six tasks. The first three of these tasks were also used during paper prototyping and heuristic evaluation. The fourth was added to evaluate the addition of the zoom out icon on notes, and the fifth and sixth were added to evaluate trash and undo.

1) Create a new bill and place it near the other bills on the desktop.
2) Create a new group that contains the 5 bills on the desktop, and sort it by due date.
3) Search for a note that contains the word “gift” and print the one that has the gift idea for Mom’s birthday.
4) Change the information for the bill created earlier.
5) Move the bill out of the “Bills” group. Delete the group, then undelete it, and move the bill back in.
6) If the user accomplished task 5 by using the trash, use undo instead. If the user accomplished task 5 by using undo, use the trash instead.

Interview.
Following the tasks, the facilitator conducted an open-ended interview with the subject. Subjects were asked what they found easy and difficult about Disorganizer, what they found obvious and obscure, and whether they would use Disorganizer to keep track of personal notes. Subjects were asked to compare the benefits and drawbacks of Disorganizer to existing systems.

In addition, we asked questions about the marquee selection process, which had proved to be problematic during earlier evaluations.

RESULTS

Overall, users found Disorganizer very simple and usable. All users experienced some issues, but were able to recover from them, and two of the three continued playing with the application after testing was done.

“It was easy and self-explanatory. Clicking on things revealed their behavior very quickly.”

“I usually get frustrated but I was able to figure things out before I got too frustrated and that means I’ll keep using it.”

In particular, none of the three users expressed any difficulty with the idea of zooming. Two expressed some surprise the first time they saw it happen, but all were quickly able to learn and use the feature.

Issues.
All users easily discovered that notes could be moved by clicking and dragging on them. However, this caused a later usability issue as groups cannot be dragged in the same way. Clicking inside a group causes a marquee selection, which is used to select multiple notes. In order to drag a group, the user must click and drag on the move icon in the upper left corner. All users were able to use this feature once they found it, but all were frustrated for a short period of time.

Selection was problematic in earlier prototypes, and continued to be problematic in this version. Users may select multiple items by clicking and dragging on the desktop or a group to create a marquee selection. None of the three subjects discovered this method. This made task #2 more laborious, as instead of selecting all the bills at once, users were forced to first create an empty group, and then drag the 5 bills in one at a time. This is still an improvement over earlier versions of Disorganizer, in which it was not possible to create an empty group.

Subjects were later shown the selection mechanism, and all stated that they found it obvious once they knew it existed.
Subjects were asked if they would prefer that the cursor change to indicate this function was available, and all stated that they would find this confusing.

All three subjects were easily able to identify note types by shape and color. Two of the three subjects stated that they identified types primarily by shape, while one primarily used color. Two of the subjects, when performing a search in task #3, used the “New Note” menu as a reminder of the appearance of each note type. One subject stated that he found the arbitrary shapes used for note types slightly confusing, but could easily learn them.

The toolbar and menu bar were a consistent source of frustration during user testing. Two of the three users showed a tendency to look for all functionality in the menus and ignore the toolbar, while the third tended to look in the toolbar and ignore the menus.

Most of the serious issues during user testing were related to this confusion. Two users did not notice the “Delete” button on the toolbar, and looked in the menu instead. In both cases, they found the “Open Trash” command, and accomplished the task by dragging items into the trash instead.

All three users had difficulty noticing the search field in the toolbar. Two of the three searched manually, by looking for notes of a certain type. All three users eventually found the search field, and were able to use it easily after that.

All three subjects initially used trash rather than undo to retrieve deleted items. However, it is unclear whether any conclusions may be drawn from this, as two of the three had used the trash to delete items rather than the delete button. The third subject stated that he used trash because it did not occur to him that this application might support undo. All three users were happily surprised to find the undo feature.

One subject compared Disorganizer to the “Stickies” program on the Macintosh, and said that it offered similar benefits. Another stated that it might be better than his current paper-based system because he could set it up (using a Remote Desktop application) so he could access it at home or at work.

Some functionality that is currently in the menus should also be accessible from other places. For example, most users expected it to be possible to sort a group by clicking on something on the group itself, rather than selecting the group, and going to a menu. A future change should add a button to groups that displays a short menu of possible actions.

We plan to add a title bar to the top of groups (running between the move icon and the zoom icon). This title bar would display the group name, and the user would be able to click on it to drag the group. The short menu, mentioned above, could also be placed on this title bar.

**REFLECTIONS ON THE DESIGN PROCESS**

We found the design process to be extremely useful in discovering usability problems as early as possible. In particular the paper prototyping accurately identified one of the most serious problems very early and at minimal cost.

We did, however, discover some issues with paper prototyping, which, if we were to do it again, we could easily address and likely produce better results. Our design relies heavily on being able to click and drag on almost any object on the desktop. However, during paper prototyping, many users failed to realize that most objects could be clicked.

We hypothesize that this was because the notes on the paper prototype were simply drawn onto the desktop. A square drawn on a piece of paper may not offer adequate affordances to suggest that it can be independently moved. If we were to repeat the prototyping experiment, and instead use a separate piece of paper for each note, attached with tape to the desktop, we might produce different results.

In addition, we hypothesize that subjects during paper prototyping have difficulty thinking of their finger as a mouse. Subjects repeatedly forgot to “click” instead of doing things directly, and very few remembered that they could “double-click” or “right-click”. We suggest providing users with a mouse, unattached to a computer, with a pencil taped to it to represent the pointer.

Heuristic evaluation was also quite useful, but also requires careful preparation. In particular, our prototype for heuristic evaluation was completely missing undo functionality. Our original design plan had included undo, but the implementation was far too difficult to finish for the early prototype. However, we did not advise our evaluators of this, and as a result, we received many comments about the lack of undo. Had we told our evaluators, or even included nonfunctional undo commands in the prototype, this space might instead have been used on more useful comments.

User evaluation was a useful portion of the design process, and in general functioned quite well. The pre-testing questionnaires provided good information, and appeared to
cause minimal extra work for the users. However, they would be more useful if we had a much larger group of subjects.

We chose to gather qualitative rather than quantitative data during the user evaluation. We felt that, given such a small group of users, quantitative evaluation was extremely unlikely to yield statistically useful results. Qualitative evaluation yielded many insightful comments. However, if we were to repeat the evaluation, we might add a short quantitative assessment after the tasks were completed.

This section would contain several simple Likert scales, asking how much the user liked Disorganizer, and whether he or she would likely continue using it. This information could be useful when comparing the effects of possible interface changes.

The interview portion of the evaluation provided little additional information. In particular, it was not useful to ask users about specific changes in the interface. Either users had figured out the interface as it presently worked, and were reluctant to consider changes, or they were simply frustrated with a problem and would not suggest alternatives.

Despite this, we found the interview useful as a debriefing tool. It allowed users to vent some frustration, and all users appeared to be more satisfied with the testing process following the interview. Two users expressed a desire to continue playing with Disorganizer following the interview.

In future user evaluations, we suggest a short post-testing questionnaire, before the interview. This may prove better at obtaining specific information, and may also spur more in-depth responses during the interview.

FUTURE WORK
Disorganizer is a somewhat interesting application, with a great deal of work needed to make it truly useful. Users suggested that for them to use an application like this, it would have to be available at all times, whenever they were using their computer. Therefore, Disorganizer would likely have to integrate itself into the operating system’s desktop in some fashion.

Besides the obvious implications (making Disorganizer visible at all times), this work should include integrating it with other personal management applications, such as calendars and e-mail.

Many users expressed a desire to have notes alert them when a due date was approaching. The best way to do this is not immediately obvious. Should the alerting note be highlighted in some way? Should it be automatically zoomed when the user opens up Disorganizer?

Another obvious extension, which was originally planned, but not implemented due to time constraints, was to allow the user to create and edit their own note types.

We have considered adding tooltips, which would show a short excerpt of a note’s contents when the user leaves the mouse over the note for a few seconds. This would allow the user to quickly browse through a set of notes without having to zoom in on each.

Automatic De-cluttering.
Disorganizer implements a simple algorithm that attempts to scale the notes in a group (or on the desktop) based on how “crowded” the group is. This algorithm simply takes the ratios of the total dimensions of the items in a group to the dimensions of the group itself, and adds a nonlinear scaling factor based on the number of items in the group.

This algorithm was based on trial-and-error, with no specific assessment done afterwards. However, it may be considered a positive sign that none of the users actually mentioned that the scaling was happening, whereas in earlier versions it was very apparent to the developers when a group looked too crowded, or notes looked to small.

Future work could attempt to establish experimentally what sizes of items make a group appear cluttered, and thus produce a robust and well-tested algorithm that does this type of automatic scaling. Such work may have implications for programs attempting to do automatic interface layout.

In addition, Disorganizer has a related problem of overlapping notes. This is mitigated, but not eliminated, by the automatic scaling. Is it possible to design an algorithm that can move notes around so that a note cannot be completely hidden, but so that the user feels that the notes still have the same relative arrangement?

The Problem of Low Expectations.
Disorganizer implements some functionality that is not typical for an application of this type. For example, in most applications that would “zoom” a note to edit it, this would be a modal interaction. In Disorganizer, all squished items are still fully active. They can be moved freely, and they can be double-clicked to zoom them directly.

As another example, none of our testers discovered the undo feature independently. Each stated that he or she simply did not expect this kind of feature in this application, and thus did not look for it.

This is primarily an issue with novice users. Once each of our testers discovered these features, he or she used them easily, and said that he or she would have no trouble remembering them.

In our design, we addressed this issue primarily in two ways: first, by trying to provide appropriate affordances, and second, by making the interface as safe to explore as possible.

However, neither of these is a satisfactory solution. In the first example, the usual affordance for indicating that an item cannot currently interact would be to show a disabled
appearance. However, what affordance can we use on the squished items to show that nothing has changed?

The issue of undo is potentially more serious. We have tried to make the application safe to explore, but this feeling of safety depends on the presence of undo. The user must explore to discover undo in the first place.

Many commercial applications have addressed similar issues with a “tutorial” or “demo” mode. In some applications, this mode is used the first time a user runs the program. However, users are known not to read manuals. Is it clear that a user would follow through this kind of tutorial? And is it worth the potential annoyance that it could cause?

CONCLUSIONS
We believe that we have shown Disorganizer to be a relatively successful outcome of the design process. The repeated evaluations picked out many subtle problems with our original design.

The use of zooming, and the heavy reliance on spatial orientation, shape, and color to distinguish items performed well during user evaluations. However, more testing would be needed to determine how well the approach scales to larger sets of data.

REFERENCES


