Menu Search Optimization

Mehrad Hassanpour

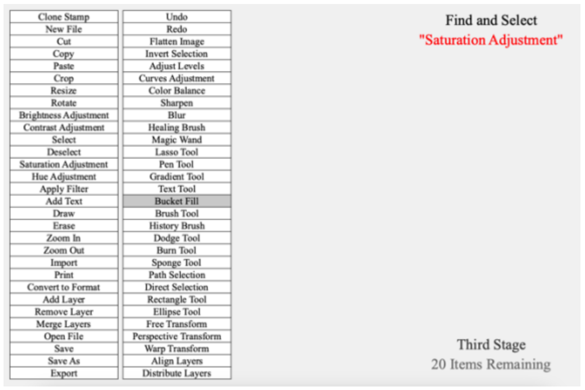
ASE205

Interaction problem

Given a set of many items, what is the most efficient way to organize them in a menu with regards to Wide vs Deep organization – Does a wider menu with a lot of items in few levels perform better or a deeper menu with few items in each level but having many levels (flat vs nested). In the final version, two separate sizes of menus are tested. One having 30 items and the other having 60 items, introducing various depths to each one and evaluating the effect of depth on each one.

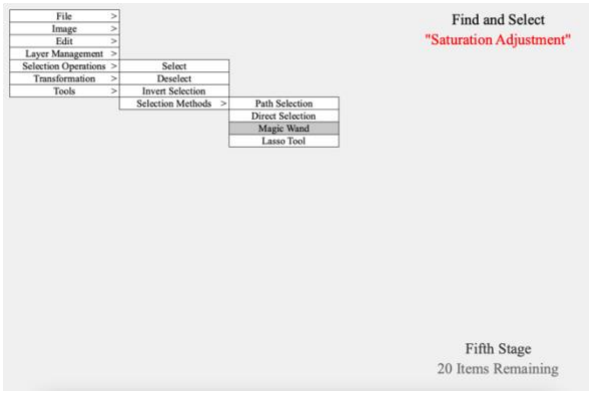
Interaction Techniques

Implement a menu system that has five different combinations – starting from a fully flat menu to a deeply nested menu. Each level in the menu will be a long column and by hovering over each item, the corresponding level will appear to the right of the item starting from the same position as that item. Hovering over the item will only expand it if there is an arrow indicating the possibility on the right side of the item. This makes it easier for the user to know if an item can be expanded or not.



The five combinations are the following:

1. fully flat organization with no nesting - 30 clickable elements
2. most items expanding to a second level - 30 clickable elements
3. fully flat organization with no nesting – 60 clickable elements
4. Most items expanding to a second level – 60 clickable elements.
5. Most items expanding to two and three levels - 60 clickable elements.



The final organization of the menus are a bit altered to be able to draw comparisons between introducing depth to different sizes of menus (30 vs 60).

There will be a prompt showing the user what item to select next. Whenever the user selects the right item, the prompt will change to the next item in the list of trials. The trials will be a set of randomly generated items.

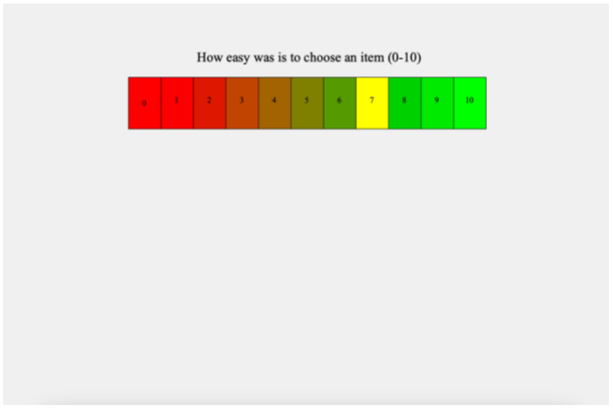
Each trial will be showing in the console:
- Selected Item ID
- Time spent in milliseconds.

And also the number of the current stage is shown at the start of each stage's output, plus at the end the user satisfaction score is shown as a single digit.

The results for each combination will be graphed and compared. Note that the trial target cannot be an item which is expandable. So, in other words, the targets are the leaves of the tree.

NOTE: my intention is to run 20 trials for each combination.

There are five stages, and the entire program is an **experiment state machine**, guiding the participant towards the end with explanations. After each stage, the user is asked to give their ease-of-use rating from 0 to 10.



I chose 30 items for the first two stages and 60 items for the other stages as a means to see the effectiveness of introducing depth for different menus sizes. My initial assumption was that the bigger menu size will benefit more from depth and the deeper the better up until a certain point, but definitely before reaching four layers.

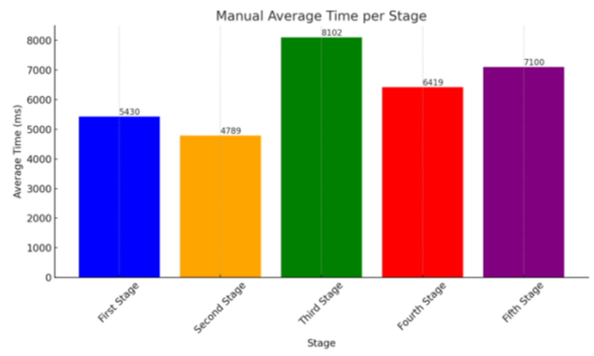
Technical overview

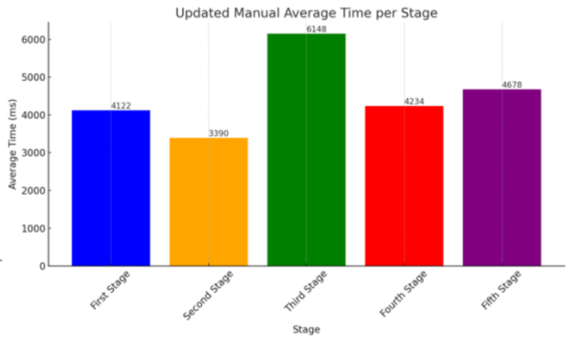
There are four major classes:
- Menu

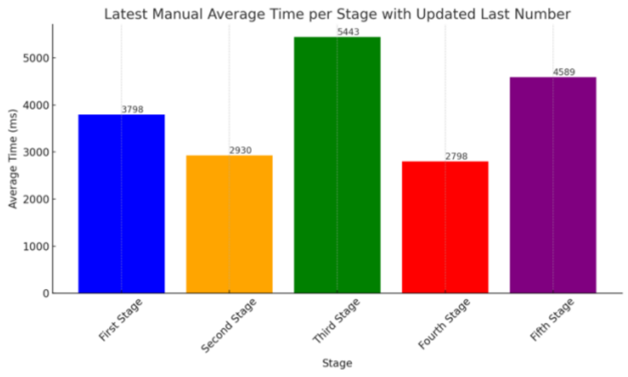
    - This class is a representation of a single layer of a menu containing many menu items. most of the operations and get methods from the menus are controlled within this class.
- Menu Item
    - This class represents each individual menu item which can be clicked on or hovered over and if the menu is expandible, will expand a menu.
- Prompt
    - This is responsible for representing the prompt that is shown to the user which has a name attribute.
- UserSat
    - This is the class that displays and captures the user's ease of use score after each stage.

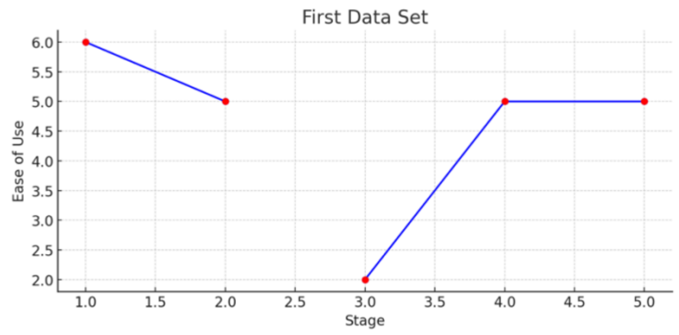
Since this program uses experiment state machine, there is a mode that advances as the user finishes the tasks. The current stage is a global variable that can take many possible stages including each combination, user ease of use score, and before and after informative stages.
The draw function of the processing keeps drawing based on which stage we are in right now.
Most of the user input like mouse clicks are directed to the menus and if the menu is expanded, the event is propagated to the next menu to have the proper effect on the target menu item.
The trial items are refreshed and randomized each time the stage is over, and are mainted by two methods of refreshTrialItems() refreshTrialItems2() where the second one is responsible for managing the 60 element stages and the first one for the 30 element stages.
After each trail end, the end of trial is triggered provided that the user has clicked on the correct menu item.

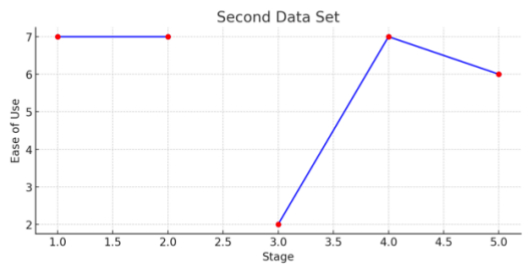
Evaluation

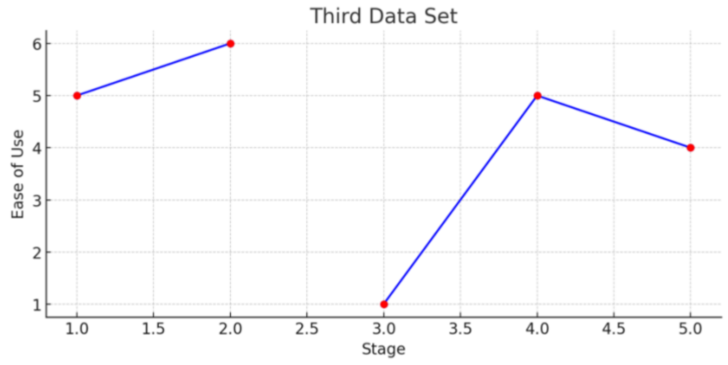












According to the gathered data from three participants, in two categories of user ease of use and performance, these observations are made:

- For the fully flat organizations, the 30 items menu took way less time compared to the 60 items menu - more visual search and clutter

- The introduction of one layer of depth to the menu, made the performance better in both menu sizes, but it seems like the effect stops when the third layer is added for the 60 items menu.
- The user ease of use data is mixed, but it is obvious that there was a jump when the user saw the stage four compared to stage three, suggesting that when the menu size is big, nesting the menus makes a better interaction. For the 30 item menus though there was not change.
- One interesting finding is that the performance increased way more when the second layer was introduced to the 60 elements menu compared to the 30 elements menu, suggesting that for big menus where the visual search is harder, it is more important to add depth and adding depth has a bigger effect.
- It seems like in the fifth stage the user gets overwhelmed with the structure of the three-layered menu system, although it still performed better than stage 3, it is worse than stage 4.

Conclusions

The important take away from this experiment was that when dealing with bigger menus, the importance of having more structure and depth is higher. But as the menu size gets smaller, it seems like introducing depth has less of an effect on the performance. Also, it is good to note that, the user can get overwhelmed with too many layers and the depth of the menu should match its complexity and size. The users have a smoother interaction with the menu when it is organized, especially for bigger menus with many menu items.

https://git.cs.usask.ca/ase205/menu-search-optimization