Comparative Analysis of Deep vs Wide Menu Structures

Arman Heydari

NSID: nki179

Interaction problem

The domain of this study is user interaction with hierarchical menu systems, a common element in many digital interfaces. The problem arises when determining the optimal structure for these menus - specifically, whether a ‘wide’ menu with many items per level, or a ‘deep’ menu with many levels but fewer items, provides a better user experience. This becomes increasingly complex with the addition of numerous items and categories It’s tricky to find the right balance. Plus, as people use the system more, they get better at finding things, so we need to consider that too. We’re going to explore these issues to help make menus easier to use. This study aims to explore these issues in depth, providing valuable insights for the design of more intuitive and user-friendly menu systems.

Interaction technique

We will use a hands-on approach to explore this problem. We plan to have about 10 people, from my friends and families, all with different levels of experience using computer menus, and different level of English proficiency to use our system. These users will be asked to search through menus of varying depths to find specific items. The order of menu depths will be mixed up, not following a set pattern. This is because as users’ complete tasks, they get better at finding items, which can affect the results. By comparing how quickly and accurately these users can find items under different conditions, we can gain insights into the ‘wide’ vs ‘deep’ menu problem. When we talk about how ‘quickly’ users find items, we’re referring to the speed at which they navigate to the desired item. ‘Accuracy’, on the other hand, is determined by the number of incorrect clicks made while trying to find the item. To account for any mistakes made during navigation, the menu system will include a ‘back’ button. This allows users to correct their path and return to the previous menu.

Tools and resources

I will use Java as the programming language and JavaFX is a powerful tool for building interactive menus. It provides a rich set of UI components, including various menu types, with customization ability. I will use IntelliJ as my IDE which is great for Java programming and GitHub for version control. The hardware needed is just a standard computer with enough processing power to run Java and JavaFX.

There are some options for datasets that I can use, and I have to choose between them while running and diving into the problem. The Kaggle restaurant menu gives me some restaurants with different food sections each with some items, in a CSV file. I can use this for designing menus with different widths and depths. Another one is the Kaggle Icons which is an image dataset of various possible icons for a phone, desktop, or tablet. Consists of 10,000 images belonging to 50 classes of icons (e.g., people, food, activities, places, objects, symbols, etc.) collected from

different technology companies and platforms (e.g., Apple, Samsung, Google, Facebook, etc.). I can also create my own dataset, for example, some categories like sports, movies, series, foods, … which are commonly known and each with subcategories and …. Which is time-consuming but might be more suited to this comparative analysis.

Here's an example of menu levels and depth in the Kaggle restaurant dataset, when I select 64 items that are familiar for the test takers.

Level = 1:

number	Item
1	Hash and eggs
2	Greek fries
…	…
64	Coca cola

Level = 2, first menu:

Section
Foods
Beverages

Level = 2, second menus:

number	Item
1	Hash and eggs
2	Greek fries
…	…
32	chicken tenders

number	Item
1	hot chocolate milk
2	can soda
…	…
32	fresh lemonade

And for level 4 it would be the same structure. For example for 4 level, if the user wants to find thai chili wings item, have to navigate like Appetizers → International Asian →Chili flavored → thai chili wings. The first menu would have 2 options, the second one also 2 options, the third one 2 options and the last one 8 options.

It would be the same with 6 levels, where the user has only 2 options per each menu.

Outline of activities

- Implement a menu system that can test many combinations of width and depth, and have a back option at every step.

- Implement a task-based system, asking the user to search for a specific item in the menu, and then another item in a new menu. It should generate these tasks and the tasks should be reasonable.
- The system should be able to record accuracy and speed consistently and save it as a CSV file.
- Test the task-based system many times and design the system in the best interactive way.
- Ask some users to do the test after I did it in many ways and had the final version.
- Gather the results and find a pattern in accuracy and speed, with respect to the user's experience, and maybe even find a pattern for the experience.

Milestone 1

The initial milestone involves implementing the system which is designed to generate tasks for the user, along with various menus within each task. It also measures and records the time and accuracy of each task, storing this data in a CSV file.

Milestone 2

Second milestone is to find users and test the system with them, and probably modify some details, it can be the data or the tasks or something about the UI.

Risks

- Not having enough data which cause redundancy in tasks and have a negative impact on results, make it unreliable.
- Can't find enough users to work with the system, results in not having enough data to find a pattern.
- System be very simple and anyone after working once or twice, can find any item very fast in any kind of menu.
- Data loss or corruption in users' saved performance
- The balance between wide and deep menus could lead to complexity in design. Selecting good parameters as width and depth is the key to success and finding a pattern.

Evaluation

If I can find a specific pattern in depth of menus in more than half of the users, it's a reasonable implementation. And if there is a correlation with the experience of user with the system with their accuracy and speed of finding items.

The goal is to draw some charts showing the accuracy (which is measured using the hit rate of items) and also the time to find an item in different conditions means different menu sizes and levels. I will compare results for menus with 1, 2, 4, and 6 levels and also determine the number of items at each of them simply just like the restaurant example's table.

Gitlab Repository Address

As I'm a graduate student of CMPT 811, I'm not sure who should I invite to the project, but this is the link and it's public access.