Angela Lin
6.831
AS3: Analysis

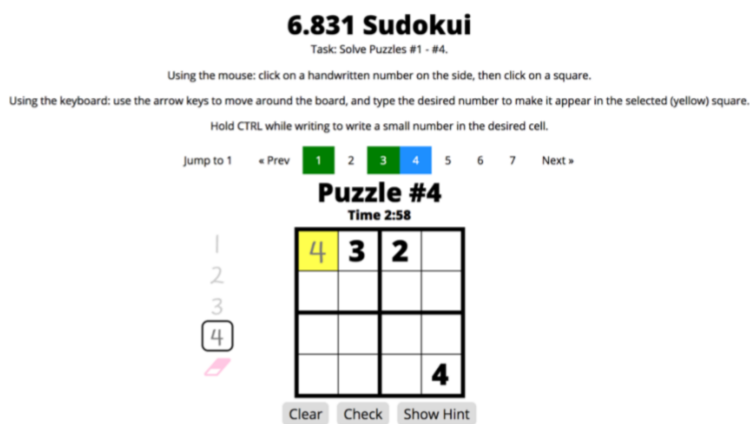
Problem

I chose to work on and improve the Sudoku UI.

Collaborators

I did not collaborate with anyone on this assignment.

Illustration



As mentioned in AS1, this illustration depicts an overview of my new interface. Users are now able to not only use the existing mouse controls, but also use the keyboard to move around and insert numbers. The highlighted square represents their current position on the board; yellow signifies that the user can edit the square, while gray signifies that the user cannot (because the number was given).

Experimental Hypothesis

Through my changes, I primarily want to improve the efficiency of completing a puzzle by improving the efficiency of *entering numbers into the Sudoku board.* By logging when users move around the board using the keyboard and when they type both valid and invalid inputs, and then comparing this data with the mouse clicks logged in version A, I can analyze how efficient it is for users to use the keyboard for control and input. In addition, by logging when and how the eraser is being used, I can analyze how often mistakes are being made, and which method users prefer to use to remove numbers.

KLM Analysis

Task[1]: *Successfully complete a puzzle as quickly as possible.*

- For interface A, users can complete the task by clicking on the number palette to select a number, and then clicking on squares on the board to draw that number.
- For interface B, users can complete the task by either using the same mouse functionalities in interface A or using the arrow keys to move around and typing in the numbers to input a specific number.

[1] I have reduced the scope of my task from AS1 (which asked users to complete Puzzles #1-4) to simplify analysis.

Baseline (Version A) KLM Analysis

First, I defined the subtasks of inserting a number because it is so frequently done. I assumed that a user would check their answers after approximately every four numbers, there was an average of 12 numbers to fill, and that the user made no mistakes. The operators used are M (Mentally Prepare), P (Point with Mouse), and B (Button press).

Subtask: Inserting number to square ("Insert"):

M, P [point to palette], BB [click number], P [point to board], BB [click square]
Total Estimated Time = M + 2P + 4B = 1.2 + 2.2 + 0.4 = 3.8 s

Thus, the full KLM analysis for Version A is:

Insert x 4, P [point to check button], BB [click check button]
Insert x 4, P [point to check button], BB [click check button]
Insert x 4, P [point to check button], BB [click check button]
Total Estimated Time = 12I + 3P + 6B = 43.2 + 3.3 + 0.6 = **47.1 s**

Version B KLM Analysis

Since in Version B, users can use the keyboard to move, I estimated that getting to a desired square would require traversing an average of 5 squares. The operators used are M (Mentally Prepare), P (Point with Mouse), B (Button press), K (Keystroke), and H (Home hands between mouse and keyboard).

Subtask: Navigating to and inserting number to square ("Insert"):

M, K x 5 [move to desired square], K [enter number]
Total Estimated Time = M + 6K = 1.2 + 1.68 = 2.88 s

Thus, the full KLM analysis for Version B is:

Insert x 4, H, P [point to check button], BB [click check button], H
Insert x 4, H, P [point to check button], BB [click check button], H
Insert x 4, H, P [point to check button], BB [click check button], H
Total Estimated Time = 12I + 3P + 6B + 6H = 34.56 + 3.3 + 0.6 + 2.4 = **40.86 s**

Dataset

For Version A, there were originally approximately ***125 unique participants***. However, not all users completed the task. Thus, I removed any users who made less than 15 actions and any users who did not ever achieve a "victory." I chose 15 as my threshold based on my KLM analysis, though admittedly, this is still a very low threshold and may not have removed all erroneous data. I then analyzed the time that each user spent; if a user played on multiple different days, I split the user's data into multiple "play sessions" based on date[2] and again removed any play sessions that did not result in victory or logged less than 15 events. Finally, I removed data when there was at least a two hour gap between actions. After sorting the data and applying these filters, I ended up with ***31 unique user sessions****.* Users collectively ***completed the task 192 times***. The ***total amount of time*** spent completing these tasks was ***32348.0 seconds*** (539.1 minutes or 8.99 hours), so each play session lasted an ***average of 1043.5 seconds*** (17.4 minutes) and each task lasted approximately ***168.5 seconds***.

For Version B, I collected data from ***14 unique participants***; even after filtering for erroneous data, I ended up with ***14 unique user sessions***; this is likely because I asked my friends to test and complete the task, and they are more likely to do as they were told and not exhibit strange behavior. Users collectively ***completed the task 93 times***. The ***total amount of time*** spent completing these tasks was

[2] Admittedly, this would have incorrectly removed any sessions that occurred near midnight. However, there were no data points near this time that would have been affected.

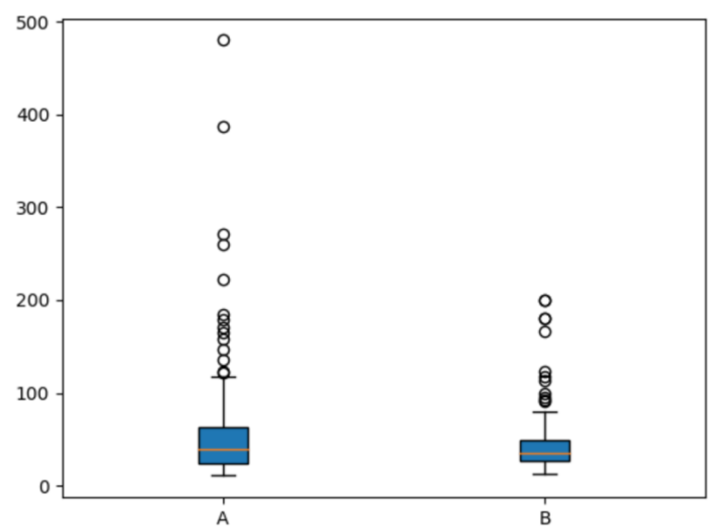
6803.0 seconds (113.4 minutes or 1.89 hours), so each play session lasted an ***average of 485.9 seconds*** (8.1 minutes) and each task lasted approximately ***73.15 seconds***.

Results

Overall, the average time spent completing the task using Version A and Version B were ***186.2 and 57.1 seconds*** each. These numbers are different from the previously reported averages because they were found by explicitly finding the start ("setup") and end time ("victory") to determine the duration of the task. These results seem to suggest that Version B is three times as efficient as Version A. However, there were outliers in the data collected for interface A that may have upwardly skewed this mean; without these outliers, the average time spent decreased to **94.8 seconds**, still suggesting that Version B was 40% more efficient**.**

The average number of clicks per task using Version A and Version B were **34.2 and 12.1,** respectively**.** The low number for Version B is likely because there were tasks that were completed without any clicks at all, since users could also input numbers by using the keyboard. The average number of key presses per task recorded for Version B was **33.9**. (It is also important to note though that there were also a few users who did not use the keyboard and relied only on the input method provided in Version A), and the average **number of squares moved before inserting a number** was approximately **1.65**; this is far lower than the number of squares I estimated in my KLM analysis (5), and may be due to how people are likely to fill out nearby squares when completing the task, as opposed to moving to constantly moving to opposite sides the board. Had I used this number (rounded to 2) in my KLM analysis instead, I would have gotten a total estimated time of 30.78 seconds for Version B, which would have suggested that Version B was 35% more efficient; this seems to align with the empirical finding of 40%.

	Average Time Per Task	Average Number of Clicks	Average Number of Key Presses
Version A	94.8 s	34.2	0
Version B	57.1 s	12.1	33.9



Box plot of the average time taken per task for Version A and Version B.
Outliers for Version A were omitted to better focus and visualize the majority of the data.
From this plot, we can see that the average time taken per task for Version B was generally always less than that for Version A.