



rijksuniversiteit
groningen

Using Users' Experience Level to Assess User Experience

Nathan John McCabe

Master Thesis - Applied Cognitive Neuroscience

S4020871

June 2023

Department of Psychology

University of Groningen

Examiner/Daily supervisor:
Sebastiaan Mathôt

A thesis is an aptitude test for students. The approval of the thesis is proof that the student has sufficient research and reporting skills to graduate, but does not guarantee the quality of the research and the results of the research as such, and the thesis is therefore not necessarily suitable to be used as an academic source to refer to. If you would like to know more about the research discussed in this thesis and any publications based on it, to which you could refer, please contact the supervisor mentioned.

Abstract

Positive User Experience (UX) predicts app use and customer retention. Accordingly, it is an useful metric by which to assess the quality of apps. For OpenSesame, an experiment-building app designed for use in psychology, neuroscience and experimental economics, this paper is the first evaluation of how it performs on quantitative and qualitative measures of UX. Since the app is used primarily by two distinct groups (students learning it, and experts using it for research), we took the opportunity to compare the two groups with the aim of identifying which components of OpenSesame are found to be intuitive by users, and which can be improved in future updates.

Using eye tracking and keypress logging can reveal when users are struggling with the interface and when they are navigating with ease (de Souza, et al., 2021; Gegenfurtner, et al., 2011; Schoemann, et al., 2020, Wang, et al., 2014). We measured: the number of clicks per minute; the average pause time between clicks; the rate of eye movements; and the average fixation duration. No significant differences were found between experts and non-experts, or across different tasks based on different components of OpenSesame. Additionally, we presented participants with a measure of perceived usability, which it performed fairly well on, and a number of open feedback questions, which contained compliments, critiques and useful suggestions. A number of possibilities for further research are discussed.

Keywords: User Experience, OpenSesame, Eye Tracking, Keypress Logging

Users' Experience Level and User Experience

Introduction

The interfaces through which technology is accessed mediate the important relationships between people and technology. The field of User Experience (UX) research concerns itself with how we use these interfaces, how we experience this use and the judgements we make about these experiences. UX refers to the perceptions a user has about the utility, ease of use, enjoyment and efficiency of a product or service (Bargas-Avila & Hornbæk, 2011; Hornbæk and Hertzum, 2017). Moreover, these perceptions are particularly relevant when they say something about the emotions a user experiences, their motivation and the degree to which they reflect on the experience during or after use (Schulze & Krömker, 2010).

With such a broad definition, researchers often disagree upon how to assess UX. Depending on the context, different measures may be more or less relevant. Using a diverse array of measures can provide a holistic overview of what a user experienced at which stage of their use journey (Renzi, 2017). When applying these many measures, positive UX scores predict app use and customer retention. This combined metric is a useful way by which to assess the quality of apps from a commercial standpoint. Beyond this, understanding what a user experiences while using a product is central to human-centred design (Wassink, et al., 2009), which puts UX as a design priority.

From a design perspective, UX research forms one step in an iterative design process. Here, the needs of a user are considered, a product is presented to them, their experience and feedback is evaluated, changes are implemented and the cycle begins again. Typically, research includes questionnaires, interviews and observations (Wassink, et al., 2009; Bargas-Avila & Hornbæk, 2011) before, during and after a presentation of a product or service. The ways of measuring UX are diverse and somewhat disagreed upon (Kim, et al., 2019; Brooke, 1996).

Questionnaires and interviews often ask product or context-specific questions, and observations normally involve a set of fixed tasks within a controlled environment. Because of the context-specificity of these questions and tasks, comparisons between studies can prove difficult. Wassink, et al., 2009, stressed the importance of “real users in the real use context,” (p18). They advocated for field studies and naturalistic observation.

Bargas-Avila & Hornbæk, 2011, conducted a review of UX research and found that design researchers often missed opportunities for deeper or more scientifically rigorous research - for example, relying on a traditional, simple emotional scale (Sidowski, et al., 1980), as opposed to more modern methods that could provide more in-depth and precise measurements of emotion. Moreover, they found an emphasis on measuring users’ affect and aesthetic appreciation for a product or service, and a lack of emphasis on context of use or how users expected a product to work. Two schools of thought were identified here: In one camp, that experience can only be assessed by first-person methods; in the other, that users’ behaviour can tell researchers more than the users can say (particularly when it comes to ratings of usability, as reported usability and recorded usability measure quite different constructs). Since this review, the use of rigorous, validated methods has increased (Arslan, et al., 2023; Hornbæk and Hertzum, 2017), and human-centred design has grown to incorporate more aspects of cognitive science.

OpenSesame

For OpenSesame (Mathôt, et al., 2011), an experiment-building app designed for use in psychology, neuroscience and experimental economics, no research has been done on which aspects of the app perform better or worse on measures of UX. It is used very often by experimenters to build virtual environments in which to run experiments. It is also taught to students, who use it for primarily coursework, dissertations and occasionally for other projects. This clear distinction between expert and novice users provides a unique opportunity for

research. In accordance with the philosophies of human-centred design, it is useful to cluster users into distinct groups that use a tool (in this case, OpenSesame) for different purposes (Wassink, et al., 2009).

Rating expertise is a difficult task. Indicators for expert performance vary depending on the domain, and experience alone is an insufficient indicator (Ericsson, et al., 2006). In our case, participants were simply asked whether they would consider themselves to be experienced with OpenSesame. Expert behaviour is defined as “consistently superior performance on a specified set of representative tasks for a domain” (Ericsson & Lehmann, 1996, p. 277). By definition, experts and novices show different behaviour when completing the same tasks (Gegenfurtner, et al., 2011; Ericsson, et al., 2006) and these differences can be used to highlight pain points - that is, commonly encountered problems that cause users to struggle with or stop a task that they are trying to successfully complete (Lamontagne, et al., 2019). In areas with intuitive, user-friendly design, differences between experts and novices are predicted to be slim. Conversely, we predict differences in behaviour to be very large at pain points. In these places, experts have learned ways to avoid or mitigate suboptimal design, but novices struggle or stop entirely. By closely comparing the behaviour of experts and novices across various tasks, we aim to detect pain points in the design of OpenSesame.

These behaviours are most convenient to compare if they are quantitatively, objectively measured, particularly if the measurement used provides insights into cognitive processes. Simultaneously, a qualitative analysis of their task completion and subjective feedback may be informative for future design decisions in revealing the different approaches and techniques experts and novices deploy to solve tasks. Together, these approaches can give a holistic view of UX in OpenSesame, as recommended by the literature (Bargas-Avila & Hornbæk, 2011; De Souza, 2021; Hornbæk & Hertzum, 2017; Kim, et al., 2019; Wassink, et al., 2009).

Variables and their Measurement

In the context of digital interfaces, certain measures of performance can reveal insights into cognitive processes. Eye movements have long been used to assess users' strategies while interfacing with computers (Goldberg & Kotval, 1999). Similarly, mouse cursors - as users' primary tool with which to interact with digital environments - are an easily-tracked embodiment of action-oriented cognition (Schoemann, et al., 2020). Furthermore, significant differences have been found between experts and novices for both eye (Gegenfurtner, et al., 2011) and mouse (De Souza, 2021) behaviour; with the latter combining both into an AI-driven analysis of mouse/eye coordination. The specific behaviours of eyes and mice can provide insights into specific aspects of performance and the thought behind it.

In regards to eye tracking, fixations can tell us a lot. Fixations are the periods of time during which the gaze remains fixed upon a specific point of interest. The eyes are relatively still, and the specific point of interest is at the centre of the visual field, allowing for detailed processing of the visual information at that point. The number of fixations and their durations can indicate searching or navigating a page, as well as attention allocation (Wang, et al., 2014), with fixation duration representing the "degree of digging into the information" (page 3). Longer fixations signify that information is difficult to comprehend, or somehow engaging. Compared to novices and intermediates, experts show slightly shorter fixation durations, slightly fewer fixations, more fixations on task-relevant areas and fewer fixations on task-redundant areas (Gegenfurtner & Säljö, 2011). We measured the number of saccades, which are changes in fixation point, per minute. This saccade rate was also inverted, which gave us the average fixation duration. We expected to replicate the findings of de Souza, et al., (2021) and Wang, et al., (2014).

By combining fixation data with mouse activity, we can focus more on action-taking as opposed to information-gathering (Wang, et al., 2014). Both are important to consider for understanding UX, but measure distinct aspects of the interface. De Souza, et al., 2021, used fixation-before-click as one of many variables fed to their AI model, which effectively clustered users according to experience level. By comparing fixation-before-click to fixation duration for all fixations, one can see how action-oriented a particular user was. Expert users are expected to gather less information, and gather information more efficiently than novice users - therefore, we can expect their proportion of action-oriented fixations to be higher than those of novices. Similarly, for tasks that are increasingly easy to use, less searching for information is needed, and the proportion of action-oriented fixations is expected to increase. This measure will not be used in this paper, but represents an interesting synthesis between eye and mouse activity.

When looking at mouse activity alone, the number of clicks per minute simply measures the frequency of interactions with the interface. This signifies a number of things: exploratory and navigational behaviour; visual scanning; speed of decision-making or the consideration of different options; engagement levels (with more clicks showing higher levels of interaction and engagement); and cognitive load, with excessive or rapid clicks indicating higher cognitive load or difficulty finding desired elements of the app. De Souza, et al., (2021), used clicks per *task* as one of a suite of mouse activity variables (distance travelled, click decision delay, path length/ideal path length). These were fed alongside eye tracking data (discussed above) into an AI model, which clustered users according to experience level. This proved useful in designing new customisation options for the website, to account for different users' needs. In our case, we simply measured clicks per minute. Inexperienced users are expected to show a higher number of clicks per minute than experts, and this difference is expected to be greater for more difficult tasks.

Using very similar data - mouse clicks over time - Arslan, et al., (2023), looked at the average pause length between interactions with an online test. Their reasoning was that any time a user is not taking action, they are performing covert cognitive processes to direct future action, like encoding, problem-solving and planning. By designing questions that weighed heavily on encoding, problem-solving and metacognitive skills respectively, they were able to assess and compare the pause lengths for each of these heuristics. They found that this worked only for questions in which test-takers attended to the content and provided an effortful attempt at solving the question. A subset of pauses were very quick, as users made snap-judgements. These rapid responses occurred more frequently towards the end of tests. Conversely, so did longer pauses caused by mind-wandering and off-task behaviour. We inverted the 'clicks per minute' data into 'minutes per click' to find values for these pauses, expecting pauses to be shorter for experts than non-experts, as they are practised with the cognitive processes typically involved in using OpenSesame.

Data gathered on both fixation and cursor *position* provides insights on where attention is being deployed (De Souza, 2021; Gegenfurtner et al., 2011). This can provide very targeted information as to which exact elements of an interface draw attention, which hold it, and which are ignored. However, this overlay of eye fixations or mouse clicks upon a display almost necessitates all tasks to happen in one unmovable page, which is not how most interfaces work. To complete almost all tasks in OpenSesame requires navigating through various windows, tabs and pages, rendering this approach too difficult to implement, without "dumbing down" the interface and thereby negating the "real-use context" that is so desirable in UX research (Wassink, et al., 2011). These measures will not be used in this paper, but instead of using this spatial data, tasks can be prepared that focus on certain aspects of the design of an interface. By analysing participants' behaviour on these different tasks, we can achieve similarly targeted information as to which aspects of the design engage users' cognitive and attentional resources.

Task Construction

Using a literature review, Renzi (2017) compiled design principles that aim to enhance a user's experience into a set of nine UX heuristics. An example of such a heuristic is design consistency, which refers to different components within the system following the same rules, responding to users' actions in the same way, independent of which component is being used. Each heuristic assesses a different aspect of user experience. As done by Souza, et al., (2021), we developed a series of tasks (**Appendix 1**) for participants to perform, incorporating features of OpenSesame that mapped onto the seven heuristics relevant for OpenSesame. The details of which heuristics map onto which tasks are included in the methods section. Different types of users (experts and non-experts) are expected to use different strategies to achieve tasks. The longer the task, the more the gap between them widens, as experts have more opportunities to use shortcuts. To mitigate this, tasks were designed to be completed in just a few steps.

Method

Participants were recruited through student whatsapp groups and an experimental psychology forum, as well as a poster at the RUG faculty of Behavioural Sciences. We recruited 7 Master's students, 8 PhD students and 4 faculty members. Of these, 10 considered themselves experienced with OpenSesame, and 9 considered themselves inexperienced. We did not record further demographic data. Experienced and inexperienced participants completed the same procedure. They were tested in one session that lasted between 30 and 60 minutes, depending on how fast tasks were completed. Each participant used the same computer in the same booth to increase data reliability.

Apparatus

The experiment took place in a small, evenly-lit windowless room, using Windows computers with 1080p monitors and Gazepoint eye trackers sampling at 60 Hz. Participants were placed 57cm away from their monitors, where they worked on tasks in OpenSesame 3.3.14, *Lentiform Loewenfeld*. A mouse- and keyboard-logging python script was used within OpenSesame (**Appendix 3**). This recorded which key was pressed, the time it was pressed, the location of each click on the screen in x/y coordinates, and the target of the click or key press (The in-app element that was clicked on or typed within). Moreover, this script logged its starting time, which allowed it to be synchronised with the eye tracking data. Simultaneously, the GazePoint eye trackers recorded eye position on the screen in x/y coordinates, which was used to identify fixations, the time between saccades, which are large, rapid eye movements.

Icon Quiz

Before participants started tasks in OpenSesame, they were presented with a list of the icons for commonly-used buttons and the question “What do you think are the functions of the following buttons?”. Their open responses were marked as correct or incorrect.

Tasks

With the keylogging and eye tracking variables being recorded, participants completed tasks in OpenSesame. They worked in the same pre-made experiment. The tasks were sorted into a freeform task and three sets of three sequential tasks. The task sets were presented to the participants in a random order, before the freeform task, which was to “Open a new file and design a simple experiment of your own.” These tasks were presented in an instruction sheet (**Appendix 1**). After each task set, the experimenter reset OpenSesame to the same example experiment and restarted the mouse- and key-logger.

By asking users of OpenSesame in an experimental psychology forum, a list of common tasks was put together. As many common tasks are dependent on previous steps, they were divided into sets. The tasks within sets needed to be completed in order, but sets were independent, which allowed us to randomise task set order and reset their progress to a baseline several times during the experiment. In Table 1, tasks are described and matched to the design heuristics (Renzi, 2017) that would be encountered in their completion. Each task has a particular heuristic that is most present, in bold.

Each heuristic is a design principle that relates to ease-of-use. ‘Place-making’ refers to the ease of self-localisation for users within a system; ‘Consistency’ refers to the need for each aspect of the program to follow the same rules and respond to actions in the same way; ‘Resilience’ refers to the flexibility of the program to cater to different users and approaches; ‘Reduction’ refers to presenting complex systems in a simple, usable way; ‘Correlation’ helps users find information more naturally, as different sources of information agree with each other; ‘Equivalency to cultural conventions’ refers to the program’s ability to understand users’ references and anticipate how they expect the program to work; Finally, content is ‘visually intuitive’ if it is easy to understand, with minimal working memory load.

Table 1

Task Descriptions and UX Heuristics

Tasks	Descriptions	UX heuristics
Set A	1 Changing the practice phase to repeat twice	Place-making , consistency, resilience, reduction, cultural convention, visually intuitive.
	2 Changing the instruction text	Place-making, consistency, resilience, correlation, cultural convention , visually intuitive.

	3	Adding two questions	Consistency, correlation, cultural convention, visually intuitive .
Set B	1	Changing criteria for feedback	Place-making, consistency, resilience, reduction , correlation, cultural convention.
	2	Adding an eye-tracking logger	Place-making , consistency, reduction, cultural convention, visually intuitive.
	3	Selecting eye-tracking software	Place-making, consistency , reduction, cultural convention, visually intuitive.
Set C	1	Running the experiment, locating a bug	Resilience , correlation, cultural convention, visually intuitive.
	2	Fixing a simple bug in a section of script	Place-making , consistency, resilience, correlation, cultural convention.
	3	Testing that the fix works	Resilience, correlation , cultural convention, visually intuitive.
Freeform		Creating an original experiment	Case-dependent

Performance on these tasks can point towards particular systems (for example, form creation) as well as particular heuristics (for example, consistency) that could be improved. After the tasks within OpenSesame were completed, participants were presented with a short questionnaire (**Appendix 2**). The first 10 items were the System Usability Scale (Brooke, 1996), followed by open questions about OpenSesame:

- What do you find most intuitive about OpenSesame?
- What do you find most frustrating about OpenSesame?
- What is the one thing you wish OpenSesame could do that it doesn't already?

- Do you have any comments or suggestions that can help us improve the user experience and interface of OpenSesame?

Analysis

The quantitative data for mouse activity - clicks per minute per task and average pause duration per task - was extracted from the keystroke log using python (generated with ChatGPT 3.5 - **Appendix 4**). In some cases, for example where participants had double-clicked “task complete”, data was cleaned manually. In the cases where they did not press ‘task complete’, the timestamp of their last keypress was used as their final time. For eye activity, saccades (changes in fixation point) per second per task and average fixation duration per task were extracted from Gazepoint data using a separate python script (generated with ChatGPT 3.5 - **Appendix 5**). For each of these four variables, an independent-samples t-test, with expertise as the grouping variable, was conducted to compare non-experts to experts on their total scores. Then, they were compared in five RM-ANOVAs across the ten task levels, with experience level as a between-subjects factor. The SUS questionnaire, conducted in qualtrics, outputs a score of perceived usability between 1 and 5. An independent-samples t-test was conducted on this, with expertise as the grouping variable.

The qualitative data analysed consisted of written feedback and the results of participants’ freeform tasks. These are summarised and discussed with the aim of identifying what separates pain points from easy-to-use aspects of the interface.

Results

Keylogging results

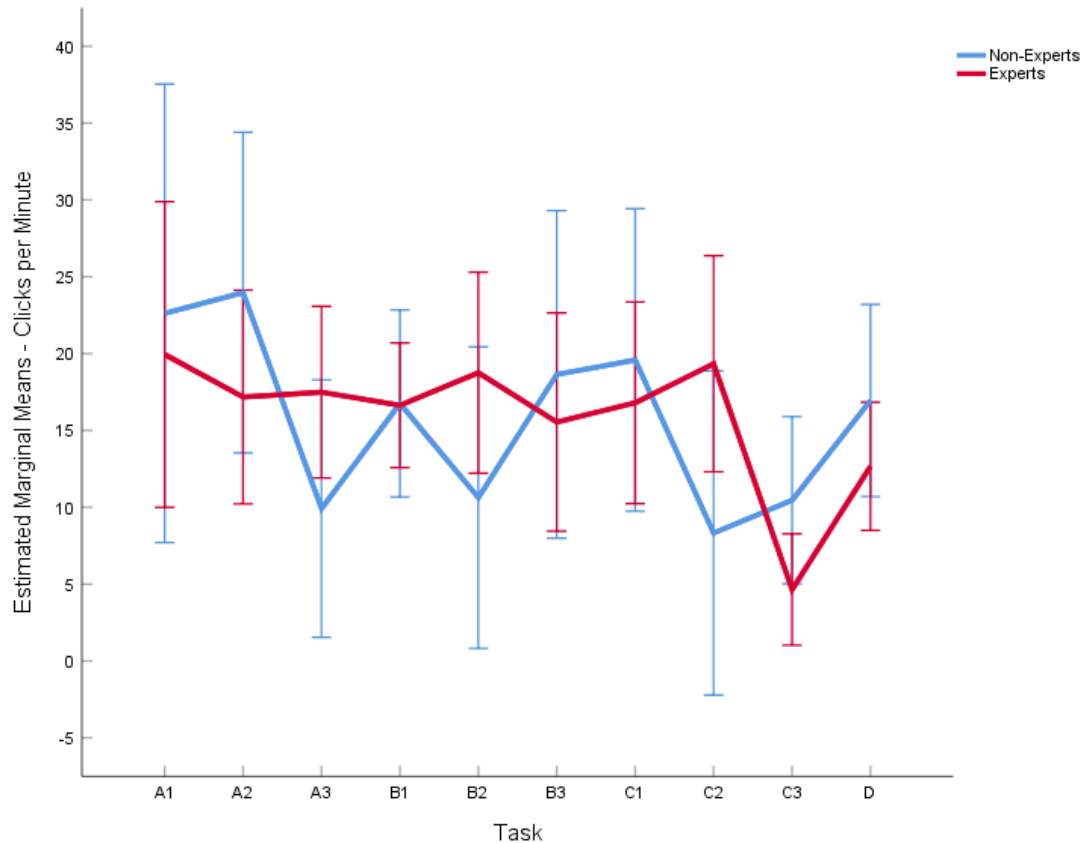
An independent samples t-test (n=19) compared the average **clicks per minute**, our measure of action-oriented behaviour, of non-experts and experts, without taking tasks into

account. The assumption of equal variances was met and there was no significant difference between the two groups: $t(17) = -0.643, p = .529$.

To examine how this differed across the ten tasks, a repeated-measures ANOVA ($n=13$) was conducted, with expertise as a between-subjects factor. Mauchly's test indicated that the assumption of sphericity was violated, $\chi^2(44) = 56.893, p = .165$. To account for this, a Greenhouse-Geisser correction ($\epsilon = .510$) was used. While there was a tendency towards a difference between tasks: $F(4.586, 50.450) = 2.288, p = .065, \eta p^2 = .172$; there was no significant interaction effect between task and expertise: $F(4.586, 50.450) = 1.507, p = .208, \eta p^2 = .120$. In other words, the relationship between clicks per minute and task did not depend upon expertise.

Figure 1

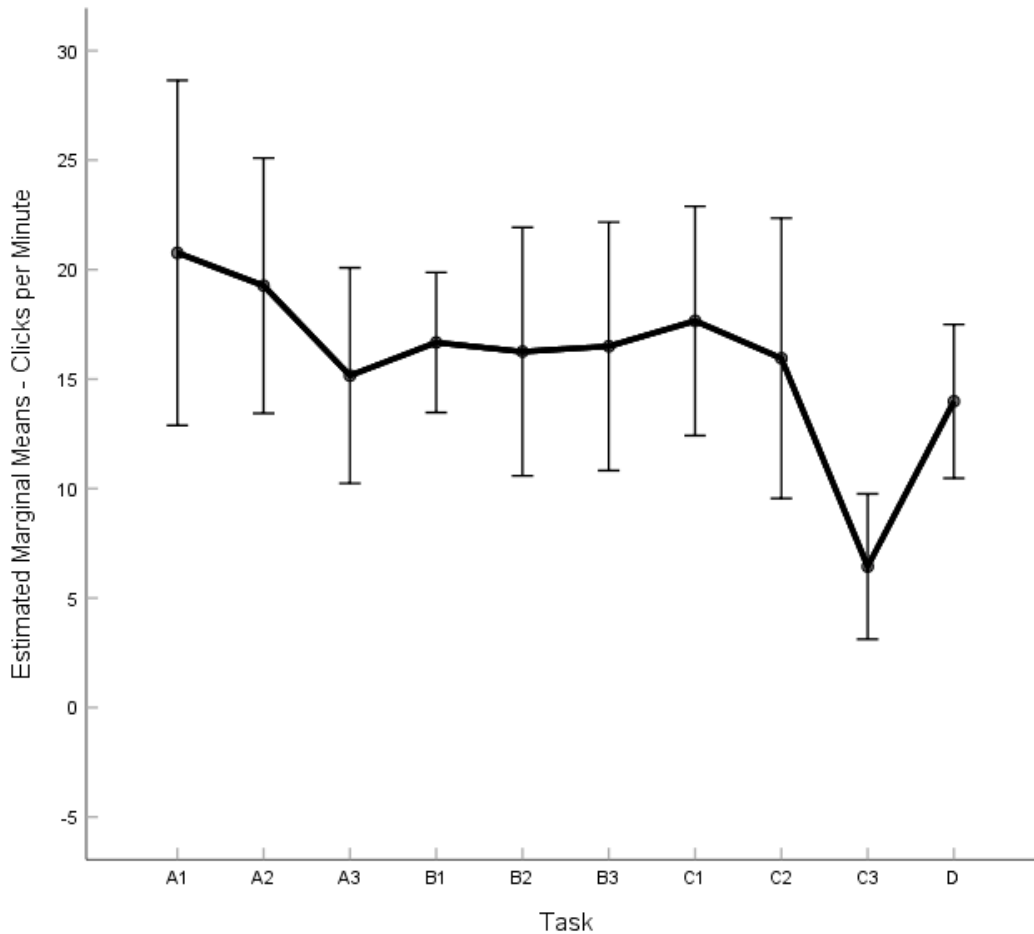
Average Clicks per Minute, per Task. Non-Experts vs. Experts.



Since the samples of non-experts and experts do not significantly differ and have similar variance, this analysis was repeated without separating by expertise, to increase the power and therefore have a closer look at how clicks per minute varied by task. Mauchly's test indicated that sphericity was still violated, $\chi^2(44) = 53.401, p = .236$. To account for this, a Greenhouse-Geisser correction ($\epsilon = .517$) was used. There was a significant difference between tasks: $F(4.656, 55.868) = 2.456, p = .048, \eta p^2 = .170$. Pairwise comparisons of estimated marginal means, with a Bonferroni correction, revealed that this difference lay only between task C3 (Test that your fix works) and task A2 (Change the instructions to say that the practice phase will repeat twice), and task C3 and task B1 (Make the experiment show feedback only after incorrect responses).

Figure 2

Average Clicks per Minute, per Task.



The same analysis was conducted on average pause duration, our measure of covert cognitive processes to direct future action. Since this is essentially the inverse of the above variable (time between clicks, as opposed to clicks/time), the relationships between variables are practically identical, but reversed. For future comparison, the mean and standard deviation were computed. For non experts, $\mu = 6.01$, $\sigma = 1.79$. For experts, $\mu = 6.13$, $\sigma = 1.29$. Since these values vary so much between systems, they are non-comparable to studies done on systems other than OpenSesame (Arslan, et al., 2023).

Eye Tracking Results

The locations that users fixated upon were collated, cleaned of outliers and sorted by expertise, into Figure 5 and Figure 6. They are fairly similar, with clustering around the menu section on the left and main work area on the right visible in both groups (Figure 6). While we expected experts to show more targeted fixations than non-experts, there is a difference: Non-experts fixated in the same spots for longer than experts, possibly showing more time spent engaging with the targets of their fixations. Figure 7 shows the interface, over which these heatmaps can be laid for context.

Figure 5

Eye Tracking Heatmap, Non-Experts

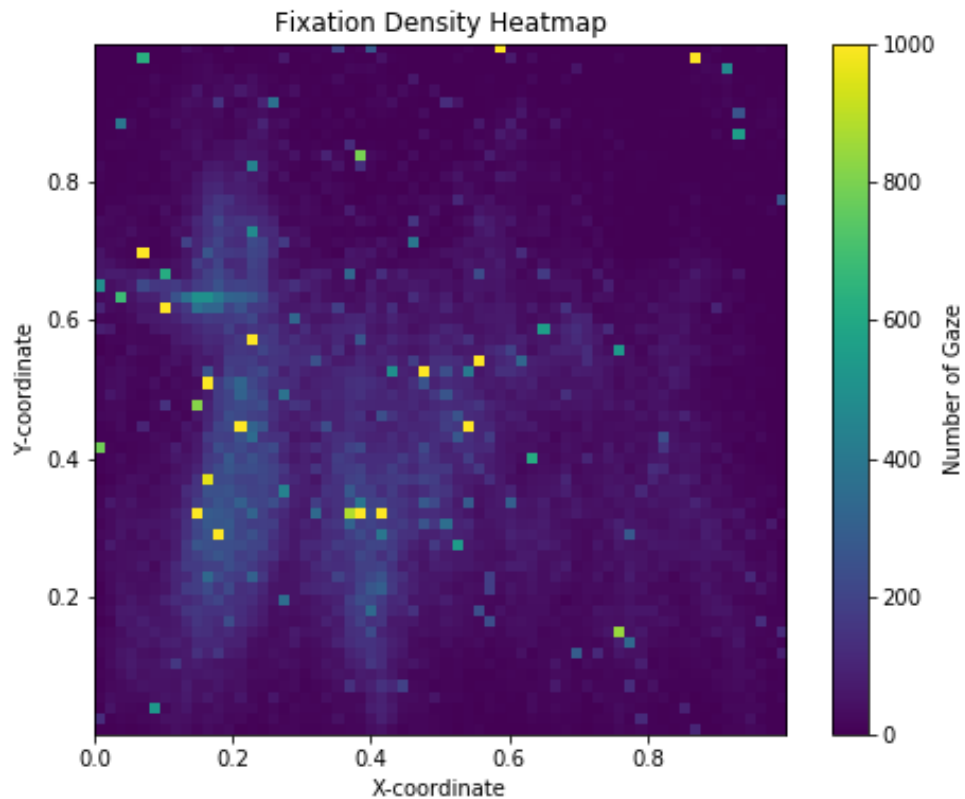


Figure 6

Eye Tracking Heatmap, Experts

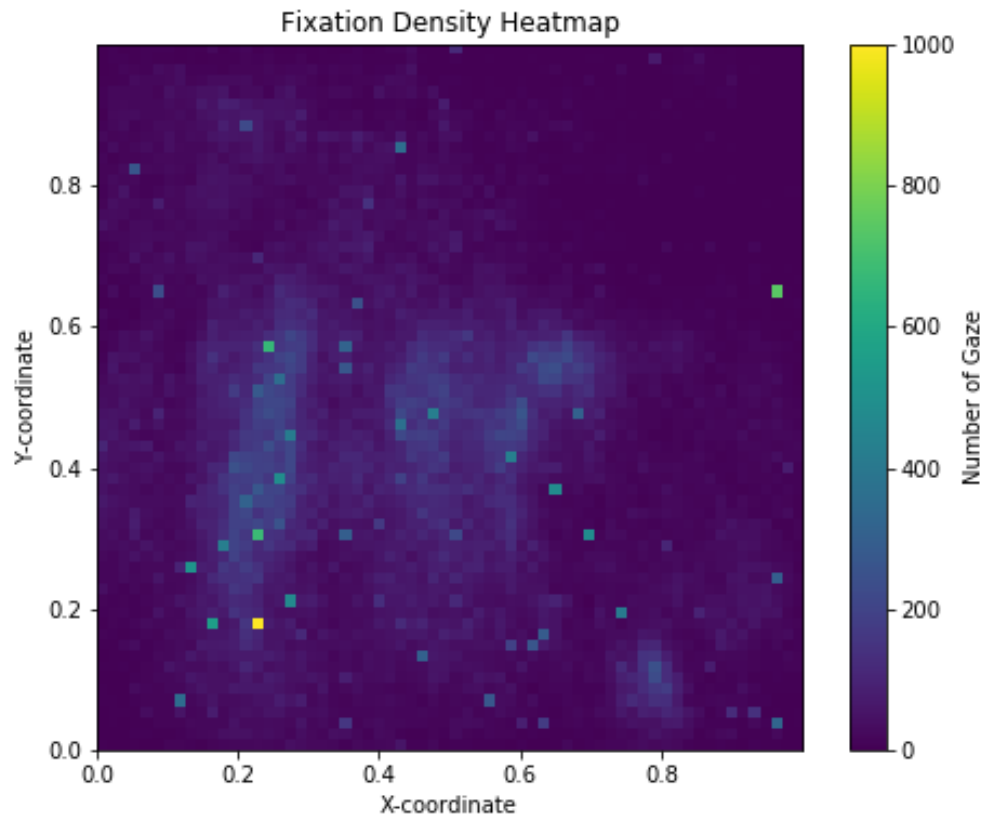
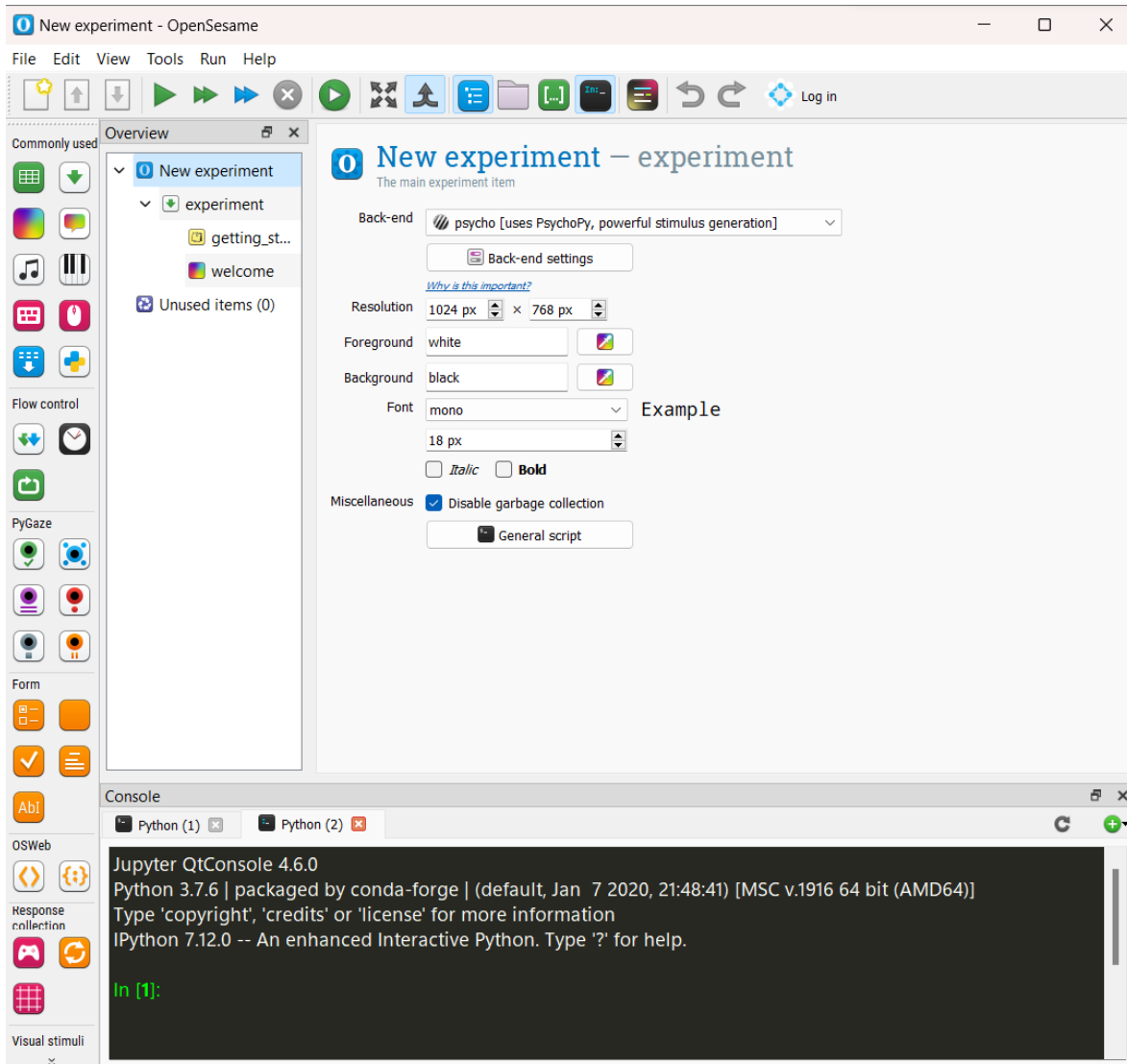


Figure 7

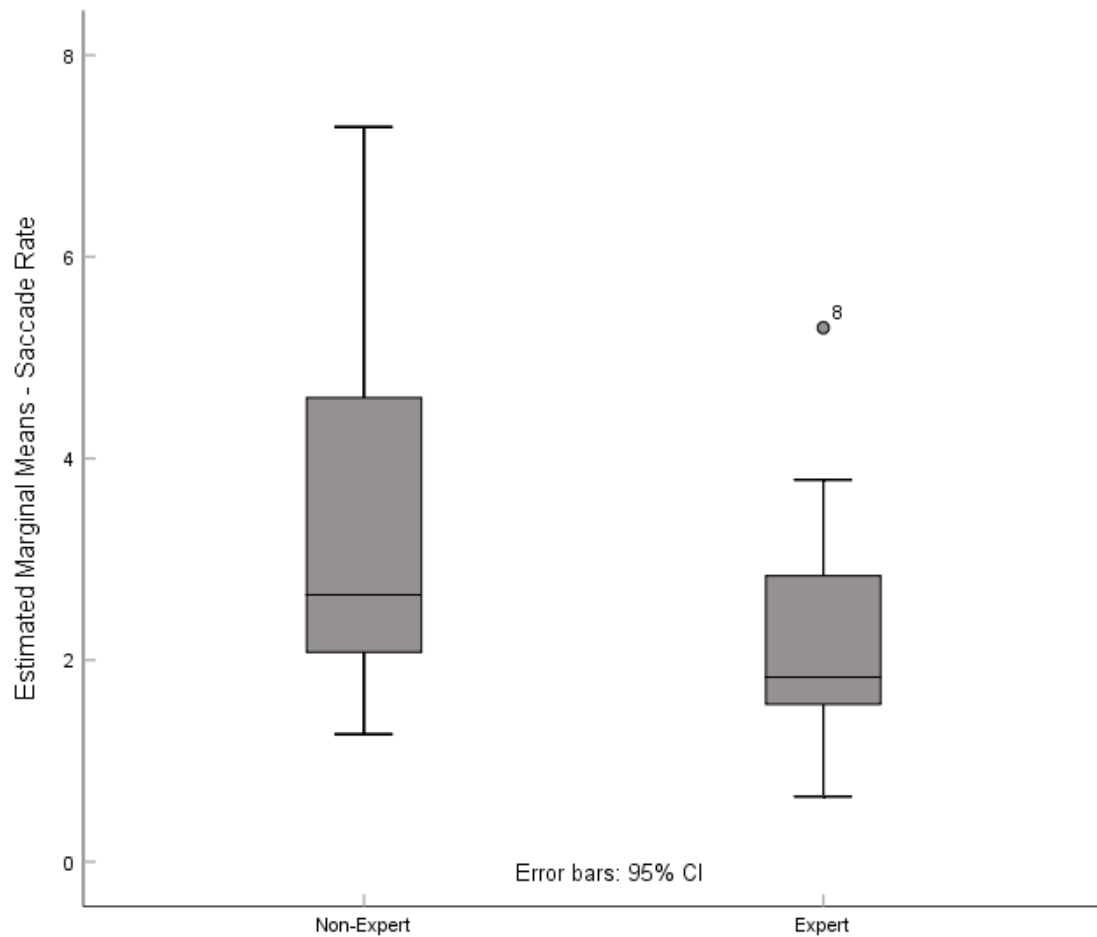
OpenSesame Interface



The **saccade rate**, or the number of fixation changes per second, was our measure of navigational and information-searching behaviour. An independent samples t-test ($n=18$) compared the average saccade rate of non-experts and experts, without taking tasks into account. The assumption of equal variances was met and there was no significant difference between the two groups: $t(16) = 1.267, p = .223$. That said, as shown in Figure 5, experts had lower variance and a lower mean, showing a non-significant trend towards experts making more efficient fixations.

Figure 8

Saccade Rate, Non-Experts vs. Experts



The **average fixation duration** is the number of seconds per fixation, our measure of how much a participant “dug in” to visual information. It is the inverse of saccade rate. As such, the relationships are very similar, with an opposite direction for the effect. The mean and standard deviation were computed. For non experts, $\mu = .39$, $\sigma = .20$. For experts, $\mu = .60$, $\sigma = .40$.

System Usability Scale and Questionnaire

Scores on the SUS were assembled into scores from 0 - 100 according to the guide in the original paper, Brooke, (1995, p. 6):

“To calculate the SUS score, first sum the score contributions from each item. Each item's score contribution will range from 0 to 4. For items 1,3,5,7,and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of SU.”

Higher scores indicated higher perceived usability. They were divided into experts and non-experts, then compared in an independent-samples t-test. While the results were not significant, ($t(17) = 1.833, p = 0.084$), experts averaged 16.69% higher than non-experts. While most individual questions tended toward a neutral response, most participants said that they would use it frequently. For the item, “I thought there was too much inconsistency in this system”, there were 4 outliers. 2 expert participants and 1 non-expert responded with a 1, 1 non-expert responded with a 5.

Figure 9

System Usability Scale. Non-Experts vs. Experts

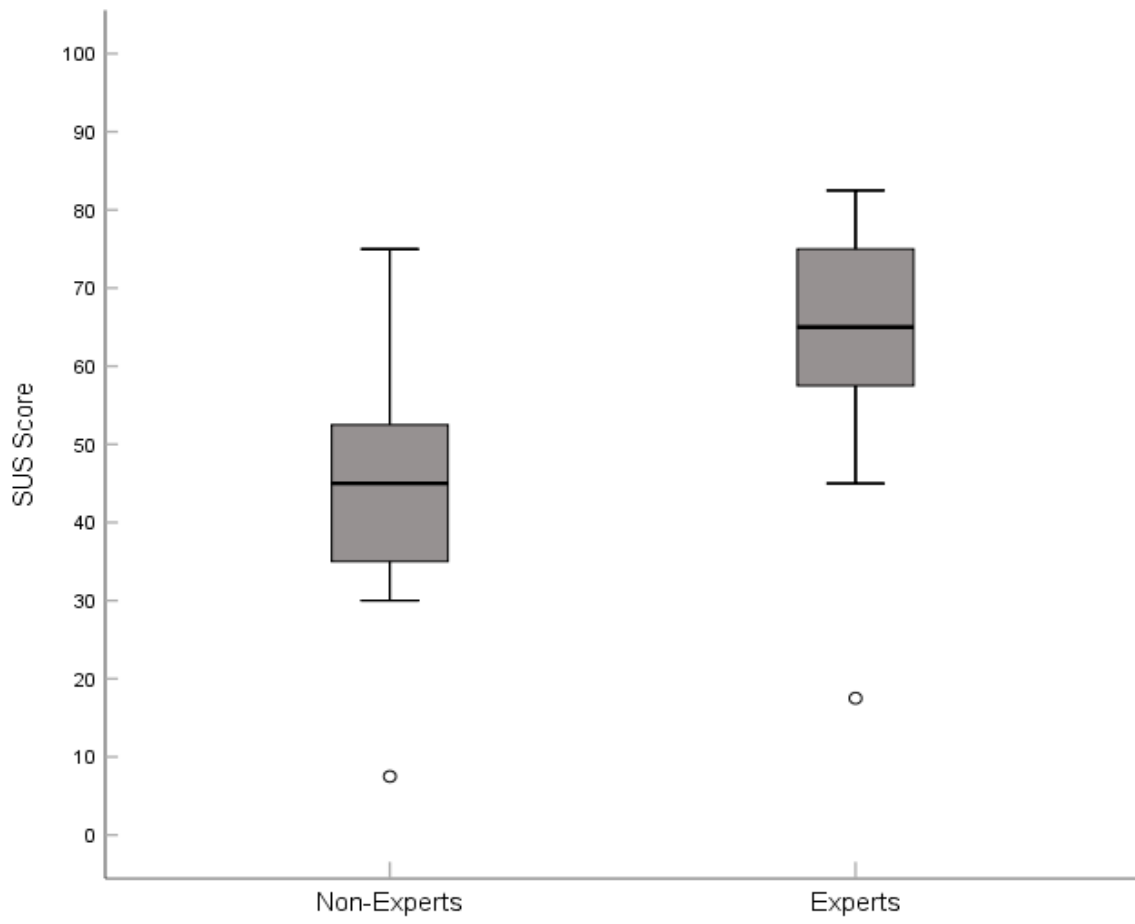


Figure 10

System Usability Scale. Individual Questions, Non-Experts.

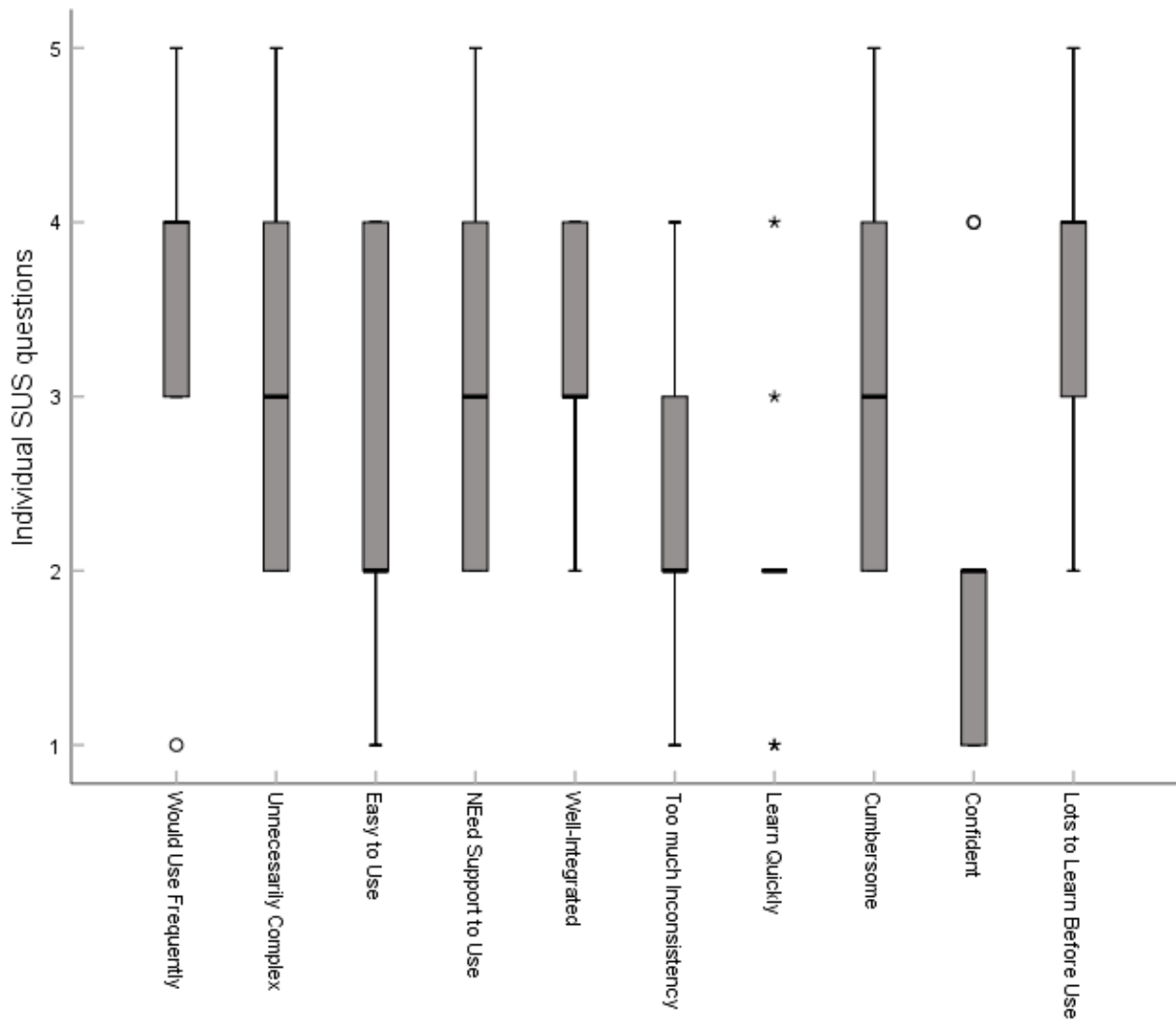
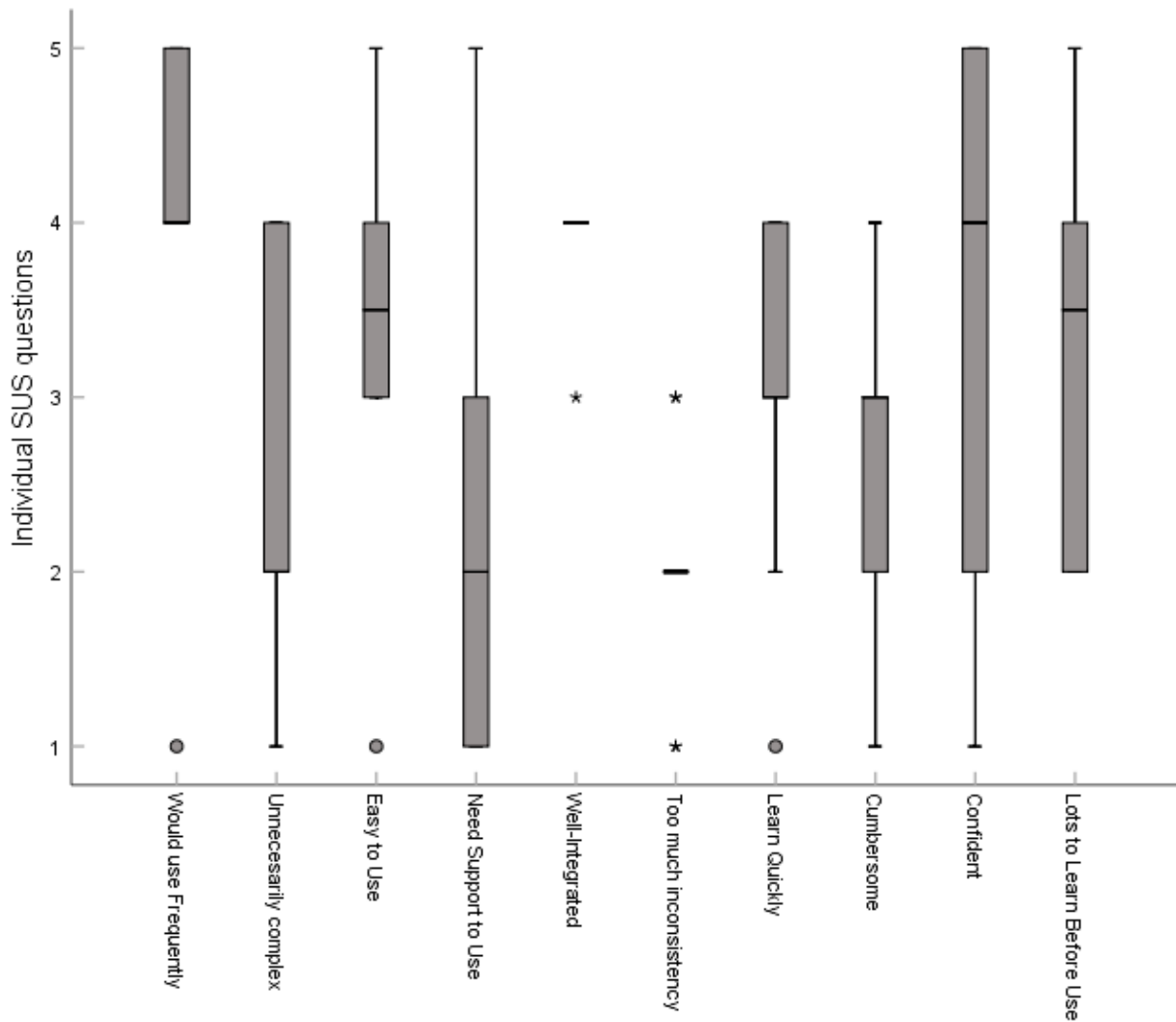


Figure 11

System Usability Scale. Individual Questions, Experts.








The questionnaire also presented participants with a list of icons that appear in the interface of OpenSesame, with the question “What do you think are the functions of the following buttons?” Their open responses were assessed as either correct or incorrect. Incorrect answers provide insights into common confusions, and are shown in the table below alongside the icons, their function, and the percentage of non-experts and experts who were correct. Of these, some were correctly interpreted by almost all participants (Console, Kill Experiment, Initiate Pygaze, Run) and some were misunderstood by almost all participants (Turn Tab into a






Dockwidget, Variable Inspector, Loop, Run in Window, Select Controls/Script View). For some icons, there were common misidentifications - For example, the Loop icon was misidentified as a Table or Spreadsheet icon by 8 of 19 participants. For other icons, there was no consensus among the participants who misidentified it - For example, One-Tab Mode was misidentified by 7 of 19 participants, but their interpretations were all distinct. This shows that in some cases, there is an alternative meaning of an icon that competes with its use in OpenSesame. In others, participants simply do not know what an icon could represent.




Table 2






Icon Understanding


Icon	Function	% Correct		Incorrect Functions	Freq.
		Non-Expert	Expert		
	Save	22%	60%	Download	6
				Move Down	6
				Decrease	1
	Open	11%	40%	Move Up	5
				Upload	5
				Share	2
				Undo	1

				Increase	1
	New	44%	60%	Favourite	4
				Marking	2
				Save	1
				Assignments left	1
				No idea	1
	One-Tab Mode/Merge Tabs	33%	70%	Increase amount	1
				Save information	1
				Insert data	1
				Random	1
				Collapse Tabs	1
				Open page	1
				Upload	1
	Turn Tab into Dockwidget	0%	10%	Full screen	8
				Resize/Expand	7
				Move around	1
				Open the display	1

	Jupyter/IPython Console	78%	100%	Insert	1
				Input field for participant	1
	Feedback Icon	22%	80%	Comment	4
				Text entry	4
				Speech	1
	Variable Inspector	0%	40%	Insert code	4
				Insert content	4
				Detailed view	2
				More	1
				Brackets	1
				No idea	1
	Kill Experiment	67%	90%	Delete	2
				Cancel	1
				Exit	1
	Logger	11%	40%	Insert data/text	5
				Download	2
				Sequence item	1

				Copy block	1
				Decrease	1
				Move group	1
				No idea	1
	Loop	22%	30%	Table/Spreadsheet	8
				Variable/Scale	2
				Trial data	1
				Block item	1
				Text entry	1
	Overview Area	55%	30%	Multiple choice	2
				List	2
				Form	2
				Instructions	1
				Log all variables	1
				No idea	1
	Initiate Pygaze*	67%	90%	Webcam	1
				Record behaviour	1

				Build something	1
				No idea	1
	Run**	89%	100%	Start a section of a product	1
	Run in Browser	32%	60%	Play/Run	8
				Video item	1
				Start the full product	1
	Run in Window	0%	50%	Quickrun	4
				Fast forward	4
				Next	3
				Re-run experiment	1
	Quick Run***	11%	70%	Fast forward	4
				Next	1
				Skip section	1
				Start experiment	1
				Freeze the product	1
	Sketchpad	22%	80%	Colour palette	6

				Layout	1
				Randomise	1
				A button to select everything	1
	Select Controls/Script View	11%	30%	Settings	4
				Switches/On-Off	3
				Toggle	2
				No idea	2
				Survey style item	1
				Dark/night mode	1

*Participants responded with “eye tracking”, two experts added “initiate”

**Answers like “play” or “run” were accepted for this button but not subsequent run buttons, which are more specific.

***Two participants that had misidentified the previous button as “quick run” said this was a quicker run.

Participants were also asked open feedback questions, (**Appendix 6**). Chat GPT-4 was used to collate and analyse their responses (**Appendix 7**). OpenSesame, generally speaking, is perceived by participants as having a user-friendly nature, thanks to its intuitive drag-and-drop interface and clear symbols. Some frustrations often arise from limitations in interface design,

difficulties with in-app coding, and navigation. For new features, users suggested multiple project views, improved tutorials, and more advanced stimulus settings. As a result, suggested improvements involve enhancing the design of the interface, providing clearer icons, and offering help resources, like tutorials.

Discussion

Interpretation of Findings

We found that there were no significant differences between non-experts and experts in saccade rate (which reveals number of fixations per minute) or average fixation duration. This is in contrast to our research (Wang, et al., 2014; Gegenfurtner & Säljö, 2011), in which experts showed more fixations targeted on task-relevant areas, slightly fewer fixations and for shorter durations. In the heatmaps, we found that experts' fixations were more distributed across the interface than non-experts, who lingered or returned to points of interest more often. For mouse tracking alone, we found no significant differences between the groups, when we expected experts to show fewer clicks per minute (De Souza, et al., 2021), and shorter pauses (Arslan, et al., 2023). The icon quiz was successful at identifying some that are less-understood than others, and showing a clear difference between experts and non-experts, which helps validate our categorization of the two groups, which was otherwise based on their own opinion of their expertise.

For differences between tasks, we aimed to identify some tasks with a larger difference between expert and inexpert mouse and eye activity. Their activity did not differ meaningfully, and the only task that significantly differed from the other tasks was Task C, "Test that your fix works". This is easily explained, as to test the fix, a participant only needs to run the experiment and click through it. They had already run the experiment in an earlier step. In doing so, they rehearsed the essential parts of this task. Data showed more clicks per minute and shorter pause

duration, which is consistent with this behaviour. All other tasks did not differ in any meaningful way. This homogeneity could be explained by longer task duration, involving more features of the OpenSesame interface - such that the behaviour averaged out, with no task exclusively involving features distinct from other tasks. Since there were no significant differences between these tasks, other than Task C, not much can be said about the design heuristics of OpenSesame, and no judgements can be made as to which heuristics the app performs well or poorly on.

While the similar data suggests that non-experts and experts interacted with OpenSesame in a similar way, this may not be the case. Uncertainty can result in more clicks and more fixations, but so can speed, which comes with experience. The two groups' user journeys may have been distinct, involving non-distinct mouse and eye activity. For example, in the Task sets A and B, there is no task to debug the experiment. Some experts nonetheless found and removed the bug early, and tested that it worked - essentially doing Task set C at the start of Task sets A and B. This extra activity, or similar activity, may have made their data similar to that of non-expert participants who engaged in inefficient work.

Our quantitative data can be broken down by question. Participants were asked what they found most intuitive, which was the user-friendly interface, drag-and-drop functionality, clear icons and symbols, experiment structure, and ease of use. Their frustrations revolved around limitations in what OpenSesame can do (single project view, screen scaling issues), loop logic and variable management, difficulties with coding and inline scripting, interface clarity and navigation. Interestingly, clarity (of interfaces, icons and symbols) was recognised as being both intuitive *and* frustrating, confirming the results of the icon quiz - that some components are very easily understood and others are barely understood by anyone, experts or non-experts alike. Later, they were prompted for comments and suggestions. Three participants suggested improvements to the icons of buttons, with one suggesting that they should be explained clearly.

When asked what features they would like to see that OpenSesame does not currently have, responses focused on being able to do more advanced, more complex tasks than the current system supports (See Appendix 6 for specifics); and on tutorial resources. It should be noted that most users of OpenSesame have been taught how to use it as part of a course. Nonetheless, in-app help buttons and tutorials would be useful. Multiple participants consulted the OpenSesame literature on the website, which currently fills this role. When prompted for comments and suggestions, participants gave similar responses - specific responses that stick out as specific and pertinent are that bugs with screen scaling should be fixed (this is elaborated on in a later section), the documentation should have better search functionality and newcomers to the app should have more guidance towards getting started.

All in all, our results show no significant difference between the mouse and eye behaviour of experts and non-experts, a slightly tighter clustering around task-essential areas for experts. The level of understanding for icons varied from extremely high to extremely low, depending on the icon. On the SUS, users tended to rate the app fairly neutrally, with no significant difference between experts and non-experts, with users saying that they would use it frequently and that there was not too much inconsistency in the system. The open feedback was generally positive, with a few useful suggestions and critiques. Some users seemed to enjoy their time in the app, others found it frustrating. In both cases, engagement ran high - as is reflected in users averaging a click every 3 seconds. Two non-expert users reported losing track of time. Participant 1 referred to this as a “kind of flow state” and participant 17 said “it didn’t feel like an hour... because it was kind of fun, actually. I like riddles. Doesn’t mean that I’m good at it, but it’s still fun and I learned something.” Participant 12 echoed this sentiment of enjoying the unfamiliar environment, saying it felt “kind of fun, but it’s challenging. I feel like I know nothing.”

There were a few specific areas for improvement that were identified in our research. First, a long item name forced it to expand in a way that couldn't be shrunk - and if it got too long, it pushed into the overview area. Second, two users hit a bug when editing text in an inline python script. The loading wheel cursor started and did not stop. This pain point was only encountered by experts - non-experts tended to stay away from the console except when explicitly asked. Third, non-experts were quite hesitant to save a file of data they didn't need; a requirement when running an experiment in the default way. This may have been a consequence of being in an experimental setting and being unsure of whether or not they were allowed to, and this minor pain point is easily negated with the quick-run feature. Fourth, the software for gaze-point eye trackers being named OpenGaze caused some confusion among users who hadn't used this particular feature before. However, those that were confused all either guessed correctly or found the answer on the OpenSesame website with relative ease.

Suggestions for Future Research

When planning the experiment, we aimed to run it on 15 non-experts and 15 experts. However, these proved difficult to find. OpenSesame users, particularly experts but also Masters students, tended to have busy schedules. Ultimately, we ran the experiment on 10 experts and 9 non-experts. The issue of low power was compounded by a number of participants with missing data. More so for the RM-ANOVAs, which necessitates removing whole participants if there is any missing data, since the dependence among repeated measurements within a participant is used to estimate within-subjects effect size and variance. Removing a case entirely ensures that all measurements are treated equally. There were a number of reasons data could be missing.

Firstly, it may not have been recorded - as was the case for Participant 9's eye tracking data. The eye tracking software had a calibration phase and a recording phase. The calibration phase wasn't ended correctly, and nothing was recorded. Secondly, it may have been written

over, as was the case for Participant 2's keylogging data. The script used to log keypresses automatically saved the datafile with the same file name, and in the case of Participant 2, this was repeatedly saved-over, resulting in data only for Task 4. Third, some participants did not press "task complete" when completing a task, resulting in big blocks of data for whole task sets, instead of individual tasks. Comparing whole task sets would have allowed comparison of these cases to all other cases, but the resulting analysis would not have been granular enough to prove insightful to UX, and it would have suffered from the homogeneity discussed above. Finally, there was an element of the experiment that completely failed: Participants were instructed to press the right control button to mark when they had given up on a difficult task. From the notes taken during the experiments, we can see that this happened occasionally with non-expert users. Unfortunately, the key-logging python script did not distinguish between right control and left control, and we were unable to identify when participants gave up. Left control is very frequently used as part of key combinations like ctrl-z. Some participants pressed right control and then "task complete", which was an on-screen button implemented in the UI. In these cases, we could tell when their task ended and isolate this data. In cases where they did not press "task complete" in addition to right control, their data was only grouped in task sets, not individual tasks. If similar research was conducted, "I give up" should be a button in the interface, as with "task complete". Discovering and resolving, when possible, these complications along the way could have been eased with the use of screen recording - which could prove useful for a number of reasons, but very time-consuming to analyse. In our case, these setbacks lead to a severely reduced number of participants' data being used in some analyses. Due to the small effect sizes we expected (de Souza, et al., 2023; Gegenfurtner & Säljö, 2011), larger power is needed to truly integrate mouse and eye activity into UX research, particularly in a "real use context" (Wassink, et al., 2011) like the one we used.

However, alternatively, simulations of particular areas of interest could provide valuable insights. A traditional mouse-tracking paradigm has the cursor start at the bottom-centre of the screen, with just two clickable response options at top-left and top-right. The speed and path of movement to the response button have both been interpreted as indicators of cognitive activity - Namely, the more a response option is activated over the course of a cognitive process (in this case, choosing a response), the more the cursor path deflects towards it. (Spivey, et al., 2005; Schoemann, et al., 2021). The curved, deflected path is then compared to the optimal straight line to create a ratio of actual path / optimal path, similar to measures of ideal search distance used by De Souza, (2021). In these scenarios, cursor speed and acceleration (in the case of nonlinear speed) should be noted, as it is often not reported in mouse tracking studies, which makes generalisation difficult. Similarly, when tracking mouse movement, sampling rate is important to getting accurate measurements of distance moved.

Traditional UX research involves semi-structured interviews and observations (Wassink, et al., 2011), with participants completing tasks under the supervision of an experimenter, who asks questions to clarify unclear behaviour. This would likely result in more actionable research than the insights found in this paper, as many variables can muddy the waters of mouse and eye activity. A measure that is often used, but wasn't in this paper due to the failure of the "give up" button, is conversion rate. This is the percentage of users who successfully complete a task (Callaghan, 2020), and very useful for determining the difficulty of various actions within an interface. With difficult or complex areas within OpenSesame identified, research could draw inspiration from Wang, et al., (2014), who looked at website complexity and its interaction with task complexity. Results were in line with cognitive load theory, with participants attending more to websites of medium complexity than to non-complex or very complex websites, when completing a complex task.

When building tasks, it is important to bear in mind that different users use different components within an app - One “expert” PhD student commented during the experiment that they were very good at some components of OpenSesame but hadn’t used other components at all. If certain user journeys - the routes they frequently take to a desired goal - were identified, tasks could be based on these and provide insights targeted to a specific use-case of OpenSesame.

To become very granular, some targets of interest may be significantly different on measures of eye and mouse behaviour, and mouseclick information (target) could be utilised to identify these outliers, revealing pain points - specifically, icons that are not quickly interpreted or recognised. After the mouse has hovered over a target for a moment, OpenSesame provides a name tag for that target. This threshold time could be used to discriminate between targets that are interpreted or recognised quickly, and those that are not interpreted or recognised without aid.

Conclusion

All in all, experts’ and non-experts’ eye and mouse activity was very similar, but their differences in skill and understanding were very clear in their knowledge of icons, their feedback and when observing them working. While the tasks (with the exception of task C) were homogenous in the measures of user activity, actionable insights into UX were gathered from responses to open questions, and comments from participants during the experiment. Despite occasional frustrations, participants tended to enjoy themselves and scored the experience somewhat positively on the SUS. Non-expert users acquired skills and knowledge about the app while working in it, and expert users completed the tasks with ease.

References

Bargas-Avila, J. A., & Hornbæk, K. (2011). Old wine in new bottles or novel challenges.

<https://doi.org/10.1145/1978942.1979336>

Callaghan, C. (2020, May). Conversion Rate Optimization and Applied UX Research. Nielsen

Norman Group. <https://www.nngroup.com/1-hour-talks/cro-ux-research/>

Ericsson, K. A., Charness, N., Feltovich, P., & Hoffman, R. R. (Eds.). (2006). Cambridge

handbook of expertise and expert performance. Cambridge: Cambridge University Press

Ericsson, K. A., & Lehmann, A. (1996). Expert and Exceptional Performance: Evidence of

Maximal Adaptation to Task Constraints. *Annual Review of Psychology*, 47(1), 273–305.

<https://doi.org/10.1146/annurev.psych.47.1.273>

Gegenfurtner, A., Lehtinen, E., & Säljö, R. (2011). Expertise Differences in the Comprehension of

Visualizations: a Meta-Analysis of Eye-Tracking Research in Professional Domains.

Educational Psychology Review, 23(4), 523–552.

<https://doi.org/10.1007/s10648-011-9174-7>

Gibbons, S. (2021, May 16). Three Levels of Pain Points in Customer Experience. Nielsen

Norman Group. <https://www.nngroup.com/articles/pain-points/>

Goldberg, J. F., & Kotval, X. P. (1999). Computer interface evaluation using eye movements:

methods and constructs. *International Journal of Industrial Ergonomics*, 24(6),

631–645. [https://doi.org/10.1016/S0169-8141\(98\)00068-7](https://doi.org/10.1016/S0169-8141(98)00068-7)

Hornbæk, K., & Hertzum, M. (2017). Technology Acceptance and User Experience. *ACM*

Transactions on Computer-Human Interaction, 24(5), 1–30.

<https://doi.org/10.1145/3127358>

Kim, Y. J., Rhiu, I., & Yun, M. H. (2019). A Systematic Review of a Virtual Reality System from

the Perspective of User Experience. *International Journal of Human-computer*

Interaction, 36(10), 893–910. <https://doi.org/10.1080/10447318.2019.1699746>

- Lamontagne, C., Sénécal, S., Fredette, M., Chen, S. W., Pourchon, R., Gaumont, Y., De Grandpré, D., & Léger, P. (2019). User Test: How Many Users Are Needed to Find the Psychophysiological Pain Points in a Journey Map? In *Advances in intelligent systems and computing*. Springer Nature. https://doi.org/10.1007/978-3-030-25629-6_22
- Mathôt, S., Schreij, D. & Theeuwes, J. OpenSesame: An open-source, graphical experiment builder for the social sciences. *Behav Res* 44, 314–324 (2012).
<https://doi.org/10.3758/s13428-011-0168-7>
- Renzi, A. B. (2017). UX Heuristics for Cross-Channel Interactive Scenarios. In *Lecture Notes in Computer Science* (pp. 481–491). Springer Science+Business Media.
https://doi.org/10.1007/978-3-319-58634-2_35
- Schoemann, M., O’Hora, D., Dale, R., & Scherbaum, S. (2020). Using mouse cursor tracking to investigate online cognition: Preserving methodological ingenuity while moving toward reproducible science. *Psychonomic Bulletin & Review*, 28(3), 766–787.
<https://doi.org/10.3758/s13423-020-01851-3>
- Schulze, K., & Krömker, H. (2010). A framework to measure user experience of interactive online products. <https://doi.org/10.1145/1931344.1931358>
- Sidowski, J. B., Johnson, J. H., & Williams, T. A. (1980). *Technology in Mental Health Care Delivery Systems*. Greenwood.
- Spivey, M. J., Grosjean, M., & Knoblich, G. (2005). Continuous attraction toward phonological competitors. *Proceedings of the National Academy of Sciences of the United States of America*, 102(29), 10393–10398. <https://doi.org/10.1073/pnas.0503903102>
- Wassink, I., Kulyk, O., Van Dijk, E., Van Der Veer, G., & Van Der Vet, P. (2009). Applying a User-centered Approach to Interactive Visualisation Design. In *Springer eBooks* (pp. 175–199). https://doi.org/10.1007/978-1-84800-269-2_8

Appendix 1

Task Form

Once you have completed each task, **press the “Task Complete” button**. This is to mark where one task ends and the next begins, so it is very important that you press it each time.



If you reach a point where you feel that you cannot complete a task, press the right Ctrl button and move on to the next task.

After each task set, let me know and I will come in to reset the experiment.

Task Set 1

- Change the practice phase to repeat twice.
- Press the “Task Complete” button.
- Change the instructions to say that the practice phase will repeat twice.
- Press the “Task Complete” button.
- After the practice phase, make the experiment ask participants if they are enjoying themselves and if they expect to do well. Their answers must be recorded.
- Press the “Task Complete” button.

Task Set 2

- Make the experiment show feedback only after incorrect responses.
- Press the “Task Complete” button.
- Make the experiment log eye tracking data for each trial.
- Press the “Task Complete” button.
- Set the experiment to use a gaze point eye tracker instead of a dummy.
- Press the “Task Complete” button.

Task Set 3

- Run the experiment. There will be a bug. Locate it and navigate to that section of script.
- Press the “Task Complete” button.
- Fix the bug.
- Press the “Task Complete” button.
- Test that your fix works.
- Press the “Task Complete” button.

Freeform Task

Open a new file and design a simple experiment of your own. There is no real objective to achieve here, we just want to see you interact with the app in a creative way.

Appendix 2

Icons

What do you think are the functions of the following buttons? (open response)



System Usability Scale

1. I think that I would like to use this system frequently
2. I found the system unnecessarily complex
3. I thought the system was easy to use
4. I think that I would need the support of a technical person to be able to use this system
5. I found the various functions in this system were well integrated
6. I thought there was too much inconsistency in this system
7. I would imagine that most people would learn to use this system very quickly
8. I found the system very cumbersome to use
9. I felt very confident using the system
10. I needed to learn a lot of things before I could get going with this system

“SUS yields a single number representing a composite measure of the overall usability of the system being studied. Note that scores for individual items are not meaningful on their own.

To calculate the SUS score, first sum the score contributions from each item. Each item's score contribution will range from 0 to 4. For items 1,3,5,7,and 9 the score contribution

is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of SU.”

(Brooke, 1995, p. 6)

Open Response

What did you find most intuitive about OpenSesame?

What did you find most frustrating about OpenSesame?

What is the one thing you wish OpenSesame could do that it doesn't already?

Do you have any comments or suggestions that can help us improve the user experience and interface of OpenSesame?

Appendix 3

```
import time

from pathlib import Path

from qtpy.QtCore import QObject

from qtpy.QtWidgets import QMessageBox, QAction

from qtpy.QtGui import QKeyEvent, QMouseEvent

class EventFilter(QObject):

    def __init__(self, parent=None):

        super().__init__(parent=parent)

        self._to = None

        self._log_path = Path('C:/Users/Nathan
McCabe/Desktop/THESIS/eventlog/eventlog.yaml')

        self._fd = self._log_path.open('w')

        self._count = 0

        self._prev_key = None
```

```
def start(self):
```

```
    self._to = time.time()
```

```
    self._fd.write(f'- to: {self._to}\n')
```

```
def accept_clicked(self):
```

```
    print('accept')
```

```
    dt = time.time() - self._to
```

```
    self._fd.write(f'''- accept:
```

```
dt: {dt}
```

```
''')
```

```
def eventFilter(self, source, event):
```

```
    if self._to is None:
```

```
        return True
```

```
    if isinstance(event, QKeyEvent) and event.type() == event.KeyPress \
```

```
        and event.key() != self._prev_key:
```

```
print(event)

dt = time.time() - self._to

self._prev_key = event.key()

self._fd.write(f"- keypress:

dt: {dt}

key: {event.key()}

target: {source}

")

elif isinstance(event, QMouseEvent) and \

    event.type() == event.MouseButtonPress:

print(event)

dt = time.time() - self._to

self._fd.write(f"- mouseclick:

dt: {dt}

button: {event.button()}

x: {event.globalPos().x()}

y: {event.globalPos().y()}

target: {source}
```

```
    ")
```

```
        self._fd.flush()
```

```
        return False
```

```
def __del__(self):
```

```
    self._fd.close()
```

```
def install_event_filter(obj):
```

```
    obj.installEventFilter(event_filter)
```

```
    for child in obj.findChildren(QObject):
```

```
        install_event_filter(child)
```

```
event_filter = EventFilter()
```

```
accept_action = QAction("Task Complete")
```

```
accept_action.triggered.connect(event_filter.accept_clicked)
```

```
opensesame.ui.toolbar_main.addAction(accept_action)
```

```
install_event_filter(opensesame)
```

```
QMessageBox.information(opensesame, 'Welcome to the experiment',
```

```
    'Press enter to start')
```

```
event_filter.start()
```


Appendix 4

```
import os
```

```
import csv
```

```
import yaml
```

```
import re
```

```
def parse_yaml(file_path):
```

```
    with open(file_path, 'r') as file:
```

```
        data = yaml.safe_load(file)
```

```
    return data
```

```
folder_path = Keylogging folder
```

```
output_file = output.csv
```

```
total_mouseclicks_list = []
```

```
mouseclicks_between_accepts_list = []
```

```
accept_dt_list = []
```

```
pattern = re.compile(r'P(\d+)Set(\w)', re.IGNORECASE)
```

```
for filename in os.listdir(folder_path):
```

```
    if filename.endswith('.yaml'):
```

```
        file_path = os.path.join(folder_path, filename)
```

```
        data = parse_yaml(file_path)
```

```
        total_mouseclicks = 0
```

```
        mouseclicks_between_accepts = []
```

```
        accept_dt_values = []
```

```
        mouseclicks_count = 0
```

```
        for item in data:
```

```
            if isinstance(item, dict):
```

```
                if "mouseclick" in item:
```

```
                    total_mouseclicks += 1
```

```
                    mouseclicks_count += 1
```

```
elif "accept" in item:
```

```
    dt = item["accept"].get("dt")
```

```
    if dt is not None:
```

```
        accept_dt_values.append(dt)
```

```
mouseclicks_between_accepts.append(mouseclicks_count)
```

```
mouseclicks_count = 0
```

```
# Handle the case where there are no "accepts"
```

```
if len(mouseclicks_between_accepts) == 0 and total_mouseclicks > 0:
```

```
    mouseclicks_between_accepts.append(total_mouseclicks)
```

```
total_mouseclicks_list.append(total_mouseclicks)
```

```
mouseclicks_between_accepts_list.append(mouseclicks_between_accepts)
```

```
accept_dt_list.append(accept_dt_values)
```

```
# Write the results to the specified CSV file path
```

```
with open(output_file, 'w', newline='') as file:
```

```
    writer = csv.writer(file)
```

```
    writer.writerow(['Participant Number', 'Task', 'Total Mouseclicks', 'Accept dt'])
```

```
    for filename, total, clicks_between, accept_dt_values in zip(os.listdir(folder_path),
total_mouseclicks_list, mouseclicks_between_accepts_list, accept_dt_list):
```

```
        match = pattern.search(filename)
```

```
        if match:
```

```
            participant_number = match.group(1)
```

```
            set_letter = match.group(2).upper()
```

```
        else:
```

```
            participant_number = '-'
```

```
            set_letter = '-'
```

```
        if len(accept_dt_values) == 0:
```

```
            writer.writerow([participant_number, f'Task{set_letter}', total, '-'])
```

```
        else:
```

```
            task_num = 1
```

```
for clicks, accept_dt in zip(clicks_between, accept_dt_values):  
  
    writer.writerow([participant_number, f'Task{set_letter}{task_num}', clicks,  
accept_dt])  
  
    task_num += 1  
  
print('Results written to', output_file)
```

Appendix 5

```
import os

from datamatrix import io

import numpy as np

import matplotlib.pyplot as plt

def extract_data(input_folder, output_file):

    # Get a list of all .tsv files in the input folder

    files = [f for f in os.listdir(input_folder) if f.endswith(".tsv")]

    # Process each file and write the extracted data to the output file

    with open(output_file, 'w') as f_out:

        for file in files:

            # Read the eye-movement data

            file_path = os.path.join(input_folder, file)

            dm = io.readtxt(file_path, delimiter='\t')

            # Select samples within the screen boundaries
```

```
dm = dm[dm.BPOGX >= 0]

dm = dm[dm.BPOGX <= 1]

dm = dm[dm.BPOGY >= 0]

dm = dm[dm.BPOGY <= 1]

# Plot the saccade trajectory

plt.plot(dm.BPOGX, dm.BPOGY)

plt.savefig(file_path + "_saccade.png")

plt.clf() # Clear the plot for the next file

# Detect saccades and calculate saccade rate

x = np.array(dm.BPOGX)

y = np.array(dm.BPOGY)

v = ((x[1:] - x[:-1])** 2 + (y[1:] - y[:-1])** 2)** .5

d = np.array(v > np.median(v) * 5, dtype=int)

n_sacc = np.sum(d[1:] - d[:-1] == 1)

duration = dm.TIME[-1] - dm.TIME[0]

sacc_rate = n_sacc / duration
```

```
# Plot fixation density as a heatmap

heatmap, xedges, yedges = np.histogram2d(dm['BPOGX'], dm['BPOGY'], bins=(16, 16))

extent = [xedges[0], xedges[-1], yedges[0], yedges[-1]]

plt.imshow(heatmap.T, extent=extent, origin='lower')

plt.colorbar(label='Number of Gaze')

plt.savefig(file_path + "_heatmap.png")

plt.clf() # Clear the plot for the next file

# Write the extracted data to the output file

f_out.write(f'{file},{n_sacc},{duration},{sacc_rate}\n')

print("Data extraction complete.")

# Example usage

extract_data(Data folder, output.csv")
```


Appendix 6

Responses to Open Feedback Questions

What do you find most intuitive about OpenSesame?

- The icon with the mouse over descriptions were very useful.
- Getting the overall structure of your experiment with the functions boxes that are easily to drag where you want them to be.
- pygaze initialiser and logging
- Easy to set the interaction part in the task.
- The symbols for the most commonly used items, and the overview area.
- how to start an experiment
- Creating block structures with a table
- the representation of the task as a tree
- seeing the flow of the experiment
- the drag and drop mechanic of a trial sequence
- The use of the few different form items, like the consent, or multiple choice form. And the overall structure you need to adhere whilst building experiments is quite intuitive.
- It is a fun platform, and I think once you get a grasp of how to use it, you'll improve quite quickly
- Some of the labels
- the inline script interaction with items really gives you a clear feel for what the items do
- I like the graphical user interface combined with the options to edit script
- Drag and drop, and plug and play
- the feedback and the instructions
- The play button for starting. Some icons, like a keyboard, make sense.

- Most of the pictograms are very clear and make it easier to find what I am looking for.

What do you find most frustrating about OpenSesame?

- I can't view (I think) two open projects at the same time.
- That you need to have a very specific order to make the experiment work and log all your variables, but i guess that is just how programming works.
- i do not know which button is used for what function. full screen doesnt work. running an experiment is bothersome because you have to run the whole thing when i just want to run one part at a time
- The loop logic and the factory design for variables.
- That you can't keep the area with the not so commonly used items open unless your cursor is on it
- write codes
- windows not being restricted to change in size
- looking up variable names that mean things
- The var.myVariable and the [variable] requirements always throw me off
- sometimes It is hard to deduce which variables are named in what way, making it harder to use inline code; also, the drag and drop mechanic funnily enough: it is hard to determine which items are important to come in sequence and which can just be added anywhere
- That it is sometimes hard to navigate where the coding error is in inline scripts. And that for example text/scripts aren't automatically shown both. (but the last one is easily selected by yourself ofcourse)
- Not enough guidance about functions
- Not sure how to add variables and so on.
- keeping track of the flow and the variables throughout run and prep phases

- The experiments look quite primitive when they are done. Some nice layout options for forms, for example, would be nice.
- Python code an 'Opensesame' code living in the same system simultaneously. This is not much of a problem for people that only use the UI, but for inline coders, this sometimes creates confusion / frustration (e.g., in the runif for sequences).
- the loops
- Not being able to find the code, but yeah, after you find it, it's alright. I did not see any syntax highlighting, which would be weird to code in.
- It is difficult to select and edit text. And the logger confuses me a bit.

What is the one thing you wish OpenSesame could do that it doesn't already?

- View two open projects at the same time.
- Maybe create some more shapes or something, since I just used those. But i'm not sure, can't come up with something right now
- going back !!! return button
- auditory stimuli setting more elegant and complicated.
- Can't think of anything to be honest
- better/easier forms (like qualtrics)
- cool my beer
- not really, it is pretty versatile. the "scaling" issue in windows could be fixed. also the icons on the left and the top cannot be scaled properly and are buggy visually
- better timing of TTL pulses in EEG research ;)
- Standard integration of Rapid Serial Visual Representation
- Have a help pop up explaining how the function works and why it is necessary
- A tutorial

- Pattern mask generation. That is, given a certain type of stimulus and associated parameters, create a pattern mask.
- have a good first tutorial
- Don't know all the functions and don't use it frequently. So not really a opinion. It pretty cool, but limited to power users or people that want to do simple things I feel
- This is hard to say because Opensesame is probably able to do a lot of things that I do not know of.

Do you have any comments or suggestions that can help us improve the user experience and interface of OpenSesame?

- It would be helpful to have two projects open at a time so I can mimic the structure/functions of one project to the other project while learning. Also, this input box is tiny...give me an essay box in Qualtrics.
- yes. less buttons, more uniform buttons, less windows, more explanation about what each button can do instead of where to drag it. show me where a bug is and how to fix it
- No
- Multiple sessions simultaneously on mac
- The documentaion is quite good, but could do with a better search function
- the icons are not intuitive. using custom variables could be more straightforward, navigating to the exact location where the error is by clicking a button on the error screen would be sweet
- Have users define the default file namings in the pop-up windows;
- I would like to have a less messy interface. Right now everything opens in the task bar at the top, and this makes it hard to navigate the different items or mostly bugs and errors. So overall, if it could become more clean, that would be great!

- Have the headings of the icons clearly display what the functions of these are. Have a quick crash course/cheat sheet for must-haves to make an eye experiment work and record.
- Make it more intuitive and allow help buttons and tutorials
- a variable reference: when do which vars get created, and what value do they have
- Some of the tabs are very small and difficult to work with. If you open the variable inspector, for example, it is sometimes very narrow and almost impossible to see the variables.
- In the case of many indents (e.g. nested block-like structure) it's sometimes hard to see the individual elements, even when extending the 'item-bar'. This sometimes makes it hard to edit the experiment. Maybe it could be more easily scrollable?
- explain more what the buttons mean and what their functions are
- The toggle between the code and interface view is hard to find. It would be nice to have the things you can drag in the interface or that are part of the interface are very close to the interface. Also, I found it strange that I could not find how to make the interface with the text larger as in full screen or something like that. Moreover, double clicking felt like a weird way to edit the text. I would expect the same thing to happen as in all other design programs, i.e. a dotted line appear around the 'selected text' and then I could click inside of it to edit text.
- Make the starting template longer

Appendix 7

Chat GPT-4 Analysis of Open Feedback Questions

Prompt:

I am conducting a qualitative analysis of user experience in OpenSesame, an experiment building app for psychology and neuroscience. Below are questions (the lines that start with Q:) and their answers (subsequent lines until the next question). Read the answers and summarise them for each question, reporting on any themes you find. These themes could be common experiences, emotions or attitudes that the participants expressed. Categorise the responses, per question, based on these themes. Provide specific quotes to support these identified themes. Provide a brief interpretation of what you find and suggest changes that would address these answers.

Q1: What do you find most intuitive about OpenSesame?

Themes: User-friendly interface, drag-and-drop functionality, clear icons and symbols, experiment structure, and ease of use.

Quotes:

- "Getting the overall structure of your experiment with the function boxes that are easily draggable."
- "The symbols for the most commonly used items, and the overview area."
- "Drag and drop, and plug and play."

Q2: What do you find most frustrating about OpenSesame?

Themes: Limitations (single project view, screen scaling issues), loop logic and variable management, coding errors and inline scripting, interface clarity and navigation.

Quotes:

- "I can't view (I think) two open projects at the same time."
- "The loop logic and the factory design for variables."
- "That it is sometimes hard to navigate where the coding error is in inline scripts."

Q3: What is the one thing you wish OpenSesame could do that it doesn't already?

Themes: Enhanced functionality (multiple project views, pattern mask generation), better tutorial resources, more advanced features (e.g., auditory stimuli settings.)

Quotes:

- "View two open projects at the same time."
- "Pattern mask generation."
- "A tutorial."

Q4: Do you have any comments or suggestions that can help us improve the user experience and interface of OpenSesame?

Themes: Improved interface design, better help resources, clearer icon design, fix screen scaling bugs, better documentation search, and more guidance on initial setup.

Quotes:

- "It would be helpful to have two projects open at a time."
- "The documentation is quite good, but could do with a better search function."
- "the icons are not intuitive. Using custom variables could be more straightforward."

Interpretation:

Participants generally find OpenSesame to be user-friendly, with an intuitive drag-and-drop interface and clear symbols. Frustrations tend to revolve around limitations in interface design, coding errors, and navigation. Users wish for features like multiple project views, better tutorials, and more advanced stimulus settings. Suggested improvements include better interface design, clearer icons, and improved help resources.

Recommendations:

1. Improve the interface design and navigation to address the limitations and make it more user-friendly.
2. Allow multiple project views to enable users to work on multiple experiments simultaneously.
3. Develop better tutorials and help resources to guide users, especially for the initial setup.
4. Enhance advanced functionality (e.g., auditory stimuli settings) to cater to users' diverse needs.
5. Make icons and function descriptions clearer for easier understanding and usage.