Investigating the Necessity of Delay in Marking Menu Invocation

Jay Henderson, Sylvain Malacria, Mathieu Nancel, Edward Lank

▶ **To cite this version:**

Jay Henderson, Sylvain Malacria, Mathieu Nancel, Edward Lank. Investigating the Necessity of Delay in Marking Menu Invocation. CHI 2020 - Proceedings of the Conference on Human Factors in Computing Systems, Apr 2020, Honolulu, United States. pp.13, 10.1145/3313831.3376296 . hal-02463247

HAL Id: hal-02463247

https://inria.hal.science/hal-02463247

Submitted on 31 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Investigating the Necessity of Delay in Marking Menu Invocation

Jay Henderson[1], **Sylvain Malacria**[2,3], **Mathieu Nancel**[2,3], **Edward Lank**[1,2]
[1]Cheriton School of Computer Science, University of Waterloo, Canada,
[2]Inria, France, [3]Univ. Lille, UMR 9189 - CRIStAL, France
{jay.henderson, lank}@uwaterloo.ca, {sylvain.malacria, mathieu.nancel}@inria.fr

ABSTRACT
Delayed display of menu items is a core design component of marking menus, arguably to prevent visual distraction and foster the use of *mark mode*. We investigate these assumptions, by contrasting the original marking menu design with immediately-displayed marking menus. In three controlled experiments, we fail to reveal obvious and systematic performance or usability advantages to using delay and *mark mode*. Only in very constrained settings – after significant training and only two items to learn – did traditional marking menus show a time improvement of about 260 ms. Otherwise, we found an overall decrease in performance with delay, whether participants exhibited practiced or unpracticed behaviour. Our final study failed to demonstrate that an immediately-displayed menu interface is more visually disrupting than a delayed menu. These findings inform the costs and benefits of incorporating delay in marking menus, and motivate guidelines for situations in which its use is desirable.

Author Keywords
marking menu; delay

CCS Concepts
•**Human-centered computing** → **User studies;** *Interaction techniques;*

INTRODUCTION
Marking menus are radial menus that allow users to select a command by making a *mark* in the direction of the desired menu item [20, 24]. Marking menus rely on two main modes of command selection. A *menu mode*, where the user makes a selection by pressing and holding down an input device, waiting for a certain delay (called *press-and-wait*) to trigger the menu display, dragging their cursor to the desired command in the menu and then releasing, and a *mark mode*, where the user selects a command by directly drawing the corresponding mark, without waiting, and possibly at a different scale.

One reason marking menus have been intensely studied in Human Computer Interaction [2, 3, 5, 30, 32, 36, 38, 39, 40] is because they implement the principle of *rehearsal*, whereby the selections in *menu mode* act as "rehearsal" for selecting commands in *mark mode*, easing the transition from a *novice* to an *expert* level of interaction [20][1].

Alongside rehearsal, marking menus typically include a press-and-wait attribute which adds a cost (typically a $1/3$ second delay [24]) to *menu mode* to distinguish, and encourage the use of, *mark mode*. In his doctoral thesis [24], Kurtenbach rationalizes displaying the menu after a certain delay on the basis that the menu *"can be distracting"*, *"can obliterate part of the screen"* and that *"displaying the menu takes time"*. Despite the extensive study of marking menus, surprisingly the impact of delay on users has received little attention.

In this paper, we investigate the necessity of delay in marking menu appearance, and its possible accompanying issues. Our motivations for doing this are two-fold. First, the delay in *menu mode* creates a cost for novice interaction. While this delay might act as an incentive to use *mark mode*, it is unclear by how much it accelerates learning in real use (i.e. is penalty an effective motivation [26] in context?). Second, while users may use *mark mode* for common commands, not every command is used frequently, thus *menu mode* interaction remains necessary for many commands, even for more experienced users. To the best of our knowledge, the *menu mode* cost has never been directly measured in expert interaction. We investigate whether a no-delay marking menu really inhibits expert performance, and if so, by how much.

We report the results of three controlled experiments comparing interaction with two types of marking menus: the original DELAY marking menu [24] and a NO DELAY marking menu. The first experiment – prioritizing experimental validity – compares these two marking menus in an abstract task where participants are prompted to select commands using marking menus of different breadth and depth. The second experiment prioritizes repetition to compare these two interfaces when users have reached expert level interaction. Finally, the third experiment, more focused on marking menu use within an application, investigates whether a NO DELAY marking menu

[1]Unlike common practice in the HCI literature, we will refer to "*novice*" and "*expert*" exclusively to describe user's behavior and/or overall level of interaction, not specific modes for a given technique.

Pre-print. Accepted to CHI 2020

impacts subjective user experience when performing a visually demanding task requiring users to insert and move graphical objects in a 2D scene. Our results suggest that a NO DELAY marking menu offers significant benefit for novice use and comparable performance even when users are practiced with the menus, at the expense of a small cost for fully autonomic individual command selection. In particular, the NO DELAY menu exhibits fewer errors, similar learning rates, and does not significantly disturb users These results allow us to present a more nuanced approach to delay in marking menus, to better inform their cost/benefit trade-off, and to discuss implications for other rehearsal-based interfaces.

BACKGROUND

Related work on rehearsal-based interfaces

Marking menus [20] are an extension of traditional pie and radial menus, with a design relying on *marks* rather than target selections. Command selection can be performed in two main modes: the *menu mode* triggered with a predetermined delay ($1/3$ of a second in the original implementation), and the *mark mode* where the user simply draws the mark corresponding to the command without waiting. Because actions are similar in both modes, this design facilitates a smooth transition from *menu* to *mark* mode by simply repeating command selections – what Kurtenbach describes as the principle of rehearsal [24]. The seminal design of marking menus was not error-friendly as the user could not reveal the menu once entered in *mark mode*. Kurtenbach's proposed solution was *confirmation mode* [24], where a user can dwell for a predetermined delay and switch back to *menu mode* – allowing re-selection or a cancellation.

Marking menus have inspired a large body of related research. Considering mostly expert users at first, research has explored ways to improve the breadth and depth – allowing more items – by altering marking menus' traditional radial shape. Examples include curved lines [3] and inflection-free simple marks [39]. Each of these techniques, without question, employed the *press-and-wait* technique, with a predetermined delay in order to switch from *mark mode* to *menu mode*. Zhao and Balakrishnan [39] noted that most of the possible advantages of their technique occurred when users made selections without waiting for the menu to be displayed. Research has also explored novice user experience through Wave menus [2] and Octopocus [5]. While both of these studies aim to improve novice use, they also still utilize a *press-and-wait* technique prior to triggering their respective *menu modes* of interaction.

Bailly et al.'s [2] justification for improving novice interaction provides strong motivation and rationale for an immediately-displayed marking menu. They note that *menu mode* is unavoidable – before a user can use *mark mode* they must interact with the *menu mode*, which may deter the user entirely if it is too slow or cumbersome. Further, *menu mode* never disappears: for example, 90% of commands in Microsoft Office are rarely used [29] as cited in [2], thus even expert users will still require *menu mode*. Lastly, the seamless transition between *menu* and *mark* modes [24] necessitates their co-existence.

The principle of rehearsal used in marking menus, i.e. having two accompanying modes of interaction and a delay before menu presentation, have influenced the design of alternative command selection techniques. An example of this is FastTap [12, 16, 17], a command selection technique that displays commands in a spatially-stable grid-based overlay interface. Users can display the interface by dwelling on a button located in the bottom left corner of a screen and select a command by tapping an element of the grid without releasing the first finger – thus, making selections via a chording gesture. Once the command location is known, users can also select commands with a single two-finger tap on the interface. Similarly, MarkPad [11] and InOutPad [6] are command selection techniques for trackpads that implement the principle of rehearsal, utilizing a delay prior to displaying the menu interface.

While the above papers leverage delay, none of these papers have investigated – or even questioned – the relative costs of benefits of delay. The closest work of which we are aware is recent research by Lewis [27], which examines the effect different values of delay, between 200 and 2000ms, have on the use of mark mode in marking menus. Using a single level, 8-item marking menu, Lewis shows that longer delays increase mark mode use, but at the cost of a significant increase in error rate; they observe more than a six-fold error rate increase for the longest versus shortest delays initially, and more than double the error rate for the longest versus shortest delays after users reach practiced use. Furthermore, in analysis of selection time, Lewis only observes faster selection time for delays of 2000ms (all other delays exhibit no statistical differences).

Rationale for delay in rehearsal-based interfaces

While delay is commonly used in rehersal-based interaction techniques [3, 17, 11], the duration of the implemented delays can vary depending on the technique (typically, from 100ms [3] to 500ms [11, 40] for marking menu variants). For its part, Autodesk's Maya 2019 incorporates a delay of approximately 230ms, as determined by counting frames from cursor change on press to marking menu appearing via screen capture running at 30Hz on an Apple Macbook Pro. Implementations of other rehearsal-based techniques also employ different values for delay (e.g. FastTap used delays of 150 [12] and 200ms [17], which suggests an explicit adjustment).

In order to better understand the rationale behind the use and selected values of these delays, we conducted non-anonymous email interviews with 4 interaction designers: G. Kurtenbach (inventor of marking menus [20, 21, 22, 23, 24]), G. Bailly (who designed marking menu variants and rehearsal based selection techniques [1, 2, 3, 4]), C. Gutwin (inventor of FastTap [12, 16, 17, 25]) and E. Lecolinet (who contributed to the design of several marking menus variants [2, 3, 33, 34] and rehearsal based selection techniques [6, 11]). We sent a single e-mail to each interviewee asking the following questions:

- In your opinion, why do Marking Menus and other rehearsal-based interfaces require delay?
- In each work, how did you choose the delay length?
- Overall, what are your views on the trade-off between penalizing novice users vs. making skill acquisition hypothetically faster?
- Are you aware of any study that tested delay's impact on performance, learnability or visual disruption?

Three of these researchers answered these questions in a single e-mail, the fourth researcher sent an additional e-mail to complement his answers shortly after sending the first one, without us asking for additional details or clarifications[2].

Initial motivations for delay

As expected, the goal of Kurtenbach's initial introduction of a delay was to avoid visual disturbance. Its introduction was indeed justified *"as a means to invoke menu or mark mode"*, thus avoiding an unnecessary menu pop-up when users already know the mark of the command they wish to select. Kurtenbach also answered that once these two modes co-exist, the delay *"may encourage learning/marking the mark"*. Avoiding visual distraction and nudging the user toward expert performance were also brought up by Gutwin as motivations for two delay-separated modes in FastTap [17]. Interestingly, Lecolinet reported that the delay in MarkPad [11] is needed, but for different reasons. MarkPad is a command selection technique that leverages gestures beginning and ending at the borders of a trackpad. However, trackpads are primarily used for cursor control. Therefore, the user could "pre-activate" a menu by mistake by starting a cursor controlling movement from a border of a trackpad that, if it were not for the delay, would display the menu. Thanks to the delay, the menu is not instantly displayed and there is no visible effect. The delay is therefore used to allow the system to distinguish between cursor control and command gesture before the gesture is completed.

Delay duration and its evolution

Kurtenbach does not remember exactly how he chose the initial duration of 333 ms, and responded that it may be based on an estimate of *"human reaction time"* even though *"if you think about it this isn't a reaction time situation"*. In later commercial marking menu implementations, he realized that they could adjust its value as low as 100ms *"and still have users reliably control when a menu is displayed and when a mark is drawn"*. Bailly answered that he chose the delays for his implementations of marking menu variants based on values used by Kurtenbach in the literature. Gutwin reported that the delay in FastTap implementations was tuned to prevent menus from appearing during mark mode. Interestingly, Kurtenbach noted that with a *"very small delay, some users never used the marks"*, but also that *"many people have to be told about mark mode. They don't seem to discover it"*. Finally, Bailly mentioned that he recently started to express doubt on the necessity of having two modes distinguished through a delay when it is possible to make a technique *"rely on recognition, not necessarily on memorization"*, referring to his recent research on hotkey learning as examples [13, 31].

Previous investigations of marking menus delay

The researchers were not aware of any study specifically investigating marking menus delay's impact. However, Bailly referred to Kurtenbach's study comparing *mark* mode to *menu* mode [24] (described in the next sub-section). Gutwin and Kurtenbach mentioned a study on hotkeys, by Grossman et al. [15], notably investigating the impact of cost applied to mouse-based selection on hotkey adoption.

[2]Full transcripts of e-mails sent and received are available as additional material to this submission during the review process

Investigating the necessity of delay in marking menus

In addition to the above reflections, other researchers have begun to question the rationale for delay. In recent work, Zheng et al. [40] studied the progression from *menu* to *mark* mode using a 500 ms delay in a marking menu designed for mobile devices. They acknowledge the lack of literature to support delay invocation, but note cognitive psychology research supports the cost of waiting (cost-based interaction) as an incentive for active memory retrieval [9]. Exploring cost-based interaction specifically, Cockburn et al. [8] showed that hiding keys' labels improved gesture retrieval for ShapeWriter [19] input, but did not result in increased text-entry speed. Moreover, if cost-based interaction supports active memory retrieval, it is unclear whether memory retrieval is necessary for accurate, efficient marking menu interaction [2, 8, 10, 40].

We are aware of only two studies conducted in the early 90s by Kurtenbach and Buxton that explore the value of having a delay [24, 22], and they present somewhat conflicting information. The first is a longitudinal study in which 2 participants used a 6-item marking menu for 8 and 10 hours, respectively. Results comparing command selection time in *mark* and *menu* modes (with delay subtracted) suggested that marks are faster, but the specific rationale for the improved performance (*"the user most likely waits for the menu to appear [and] must then react to the display. [...] Thus, a mark will always be faster than menu selection, even if press-and-wait was not required to trigger the menu."* [24]) was – to the best of our knowledge – only evaluated with this limited testing based on estimates, and never contrasted with a NO DELAY condition. The second compared *menu mode* (menu always visible) with *mark mode* (menu always hidden, and only shown during the first 6s of each block), using a menu layout of numbered items labelled in clockwise order. Average execution times and error rates were significantly lower in *menu mode*. However, *mark mode* was strongly disadvantaged in this study, in spite of using a layout of ordered menu items, because participants could only consult the menu for the first 6 seconds of each block.

RATIONALE FOR TESTING NO-DELAY MARKING MENUS

Given the above research and information elicited via email from Kurtenbach and others, we find merit in examining the use of delay for *menu mode* activation. In 1991, Kurtenbach proposed that *"even if a user did not have to pause to signal for the menu to be popped up, one would still have to wait for the menu to be displayed before making a selection. In many systems, displaying the menu can be annoyingly slow and visually disturbing"*. However, neither he nor any of the other surveyed experts were aware of any studies that had specifically evaluated visual disturbance, and, with advances in computing over the past 27 years, the likelihood that displaying a menu would be "annoyingly slow" is low: as of 2019, the majority of systems display even the most complex user interfaces in milliseconds. Further, effective use of threading in GUI design can ensure that interfaces remain active even in the presence of costly computational tasks and users can act even before the menu appears, if displaying it is slow, as there has been anecdotal evidence that expert users avoid these issues by *"mousing ahead"* in pie menus [18], as cited in [20].

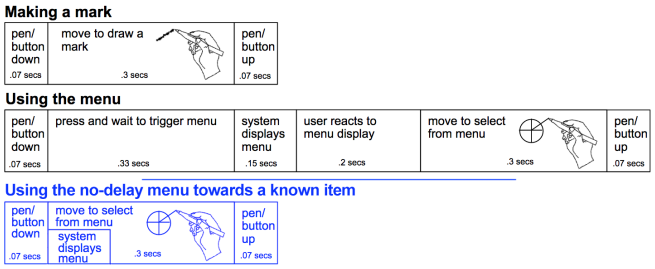

Figure 1. Top: Reproduction of figure 4.15 in [24]. Bottom: Our hypothesis on the time costs of a no-delay Marking Menu.

Delving more deeply into issues of mousing ahead and temporal costs, consider Figure 1. Figure 1-top is a reproduction from figure 4.15 in [24]; it describes users' expected behaviour when using mark mode. In the center is user's expected behaviour in menu mode, where we can see additional costs to menu mode - including system display and user search costs. However, whether there is a cost to menu display is debatable. Consider Figure 1-bottom, our representation of an alternative hypothetical time costs of a no-delay marking menu. Given that the system can capture input immediately and continuously, does the user "most likely wait for the menu" (shown as *user reacts to menu display* time in Figure 1-center) if they already know what gesture to perform? Or might the user simply begin to move, mouse ahead [20], removing the cost of menu display (bottom)? Abundant research has shown that visual memory is quick to build and robust (see [37] for an extensive review). Cockburn et al.'s model of expert performance in linear menus argues that, in practiced use, users act without visual search delay [8]. Assuming that after sufficient practice the user acquires knowledge of the visual layout, removing the delay (and therefore the *mark*-only mode) may reduce the temporal penalty without harming memorization. Other possible issues, such as distraction and occlusion, would remain to be investigated. The question then becomes whether or not we can determine the relative costs of these factors.

One could argue that, even if DELAY is not needed, it does no harm, so why study no delay? A reason to explore DELAY's benefits is, alongside potential benefits in learning and preventing visual disruption, there are potential costs to delaying the visual display of the menu:

- First, delay is a *cost* when performing command selections in *menu mode*, penalizing users if they are unfamiliar with the menu or with the specific command being invoked.
- Second, selecting a command in *mark mode* requires the user to memorize the corresponding mark beforehand, i.e. it leverages recall rather than recognition.
- Finally, with delay possibly acting as an incentive to use *mark mode*, the user may try to select commands via *mark mode* even if when not entirely sure of the correct mark, which might increase error rates even for practiced use [27].

To contrast the cost of a NO DELAY marking menu with a DELAY-based marking menu, we conducted three experiments. First, in a controlled experiment we evaluate the comparative performance of DELAY and NO DELAY marking menus by testing the following hypotheses:

H1: NO DELAY *marking menus have a lower command selection time than* DELAY *marking menus*, because of the artificial delay imposed on the *menu mode* when selecting less practiced or less frequent commands.
H2: NO DELAY *marking menus have lower error rate than with* DELAY, because visual feedback is always available.
H3: NO DELAY *marking menus are slower than with* DELAY *in highly practiced use*, because trained participants will wait for the menu to appear and will suffer visual disruption.

In our first experiment, we find little benefit of delay-based activation, possibly because our participants did not reach a sufficient level of expertise – or autonomic response – for mark mode commands. We, therefore, conduct a second study that examines the limits of expert level use to identify whether – with simulated 'perfect' expertise – we can see and quantify a benefit from DELAY.

Finally, in a third experiment we evaluate visual distraction and occlusion issues in menu mode. Evaluating these factors is challenging: the relative costs of visual distraction and disruption are subjective assessments of the user and are only present when using a marking menu to accomplish a specific task. We leverage a simple comic replication task, and elicit subjective assessments via Likert Scales and qualitative interview data to test the following hypothesis:
H4: DELAY *marking menus are, subjectively, less visually disruptive and occluding than* NO DELAY *marking menus*, because the *menu mode* need not be used during the performance of individual practiced commands.

STUDY 1: EVALUATING THE IMPACT OF DELAY

As a first step in evaluating the relative costs and benefits of DELAY versus NO DELAY, we conducted a controlled experiment to contrast the impact of delay on the performance of marking menus. Specifically, our experiment is designed to evaluation hypotheses 1, 2, and 3 in the previous section.

Experimental Procedure

Participants and Apparatus

Sixteen paid participants were recruited for the study. Average age was 23.44 (SD = 2.99). Four participants identified as female and the remaining 12 identified as male. All participants were post-secondary students from two different technically-focused universities. Each participant signed an informed consent form before starting the experiment.

The interface was displayed on a 28" ASUS PB287Q monitor with 1080p resolution. User input was achieved using a Logitech M100 mouse. We ran the application on a Macbook Pro running OS X Version 10.11.6. The application was written based on a JavaScript implementation [35] of Kurtenbach and Buxton's marking menu [20, 21, 22]. The application was modified to suit the purposes of the current research, which included, the addition of *confirmation mode*, the ability to log user behaviour and to reflect conditions outlined above.

Familiarization

Prior to completing the study, users were administered a verbal and visual demonstration of how to use marking menus within

both conditions, including all modes: *menu mode*, *confirmation mode*, and *mark mode*. The demonstration was conducted by the experimenter - thus, participants did not interact with the marking menus until the experimental task began.

Task and Stimulus

Participants were instructed to select commands with a marking menu using the right mouse button. For each trial, the participant had to select a target command (displayed on top of the window) with a MARKING MENU (DELAY or NO DELAY) of a given LAYOUT (4, 8, 4×4, or 8×8 items).

The DELAY marking menu had a 333 ms delay to enter into *menu mode* and display the menu, and a 200 ms delay to display the sub-menu[3]. In this condition, participants could also use *confirmation mode* which displayed the menu after a 333 ms delay. The NO DELAY marking menu had 0 ms delay to display the menu and the sub-menu was opened when participant entered the corresponding menu item.

For each menu and layout, participants performed 10 BLOCKS of command selection. For the 4-item configuration, participants performed 10 BLOCKS of 4 command selections (there were only four commands); for all other configurations, we selected 8 target commands and participants performed 10 BLOCKS of these 8 commands presented in a random order. We used two different command sets for each menu configuration, counterbalanced across conditions, to control for confounds of learning behaviour and confusion between categorical selections. Participants were permitted to take a break after each menu configuration was complete.

Given this task, the experiment was a $4 \times 2 \times 10$ within-subjects design, with the following factors and levels: LAYOUT (two 1-level configurations of 4 and 8 items, two 2-level configurations of 4×4 and 8×8 items. MENU (DELAY vs. NO DELAY), and BLOCKS (0 to 9). Presentation orders of MENUS and LAYOUTS were counter-balanced across participants using a Latin Square design. Therefore, in total, we collected $(1 \times 4 + 3 \times 8)$ commands $\times$ 10 blocks $\times$ 2 Menus $\times$ 16 participants $=$ 8960 selections in total.

Dependent Measures

The primary dependent measures were *Selection Time* (time from stimulus to correct command selection), *Execution Time* (time from the last mouse press to correct command selection) and *Error Rate*. Additional dependent measures were *Preparation Time* (time between the display of the target item and the first mouse-down event) and the proportion of *mark mode* usage in the DELAY condition.

Overall Time and Accuracy

In the following section, we used multi-way analyses of variance (ANOVA) for the independent variables MENU, LAYOUT, BLOCK, and their interactions. Participant is always included as a random factor using the REML procedure of the SAS JMP package. Post-hoc tests are Tukey tests when there are more than two levels.

[3]The duration of this delay is not specified in Kurtenbach's thesis [24]. We empirically set it to 200 ms to minimize accidental activation without penalizing this condition's performance.

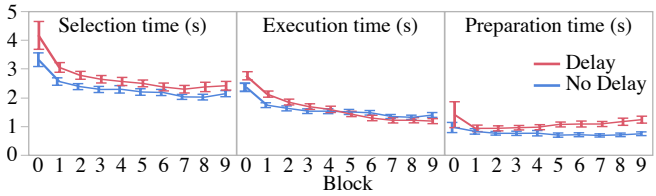

Figure 2. Average *Selection*, *Execution*, and *Preparation Times* per BLOCK and MARKING MENU condition. Error bars are 95 % CI.

We systematically removed the first trial of each participant (once in each DELAY/NO DELAY condition), which took distinctively more time than every other trial as participants were discovering the techniques. In what follows, TRIALS are ordered from 1 to 40 for 4-menu-item LAYOUT and from 1 to 80 for all other LAYOUTS (see subsection *Task and Stimulus*), and represent the ordered trial indexes for a given participant in a given MENU and LAYOUT condition. BLOCKS are ordered groups of trials containing exactly one selection of each target of a MENU $\times$ LAYOUT condition: 4 trials for LAYOUT 4×4, 8 for the other LAYOUTS. We present our analysis for each of our three hypotheses in turn in the remainder of this section.

H1: Menu Selection Time

Our first hypothesis posits that selection time is shorter, overall, for NO DELAY marking menus because of the cost imposed on less practiced use. To assess this, the different time measures were aggregated as medians instead of means to discard outliers and account for asymmetric distributions; residuals were found to follow a normal distribution. Time results are shown in Fig. 2.

To evaluate H1, we analyze *Selection Time* against MENU, LAYOUT, and BLOCK. Regarding median *Selection Time*, we found a significant effect of MENU ($F_{1,1185} = 76.29$, $p = .0001$), LAYOUT ($F_{3,1185} = 623.43$, $p = .0001$), and BLOCK ($F_{9,1185} = 63.01$, $p = .0001$), as well as significant interaction effects: MARKING MENU×LAYOUT ($F_{3,1185} = 3.76$, $p = .05$), MARKING MENU×BLOCK ($F_{9,1185} = 2.58$, $p = .01$), and LAYOUT×BLOCK ($F_{27,1185} = 7.42$, $p = .0001$).

Post-hoc tests revealed that the DELAY condition (mean 2294 ms) was significantly slower than the NO DELAY condition (2042 ms). This allows us to reject H1's null hypothesis and claim that our data support improved performance overall for NO DELAY marking menus. Our results also not only revealed that *Selection Time* was longer for 1-level marking menus than for 2-level ones, but also that all LAYOUT levels are statistically different from each other: 4 (1393 ms) $\ll$ 8 (1777 ms) $\ll$ 4×4 (2512 ms) $\ll$ 8×8 (2990 ms). Overall, as expected, our results also revealed that *Selection Time* decreased with BLOCKS at a decelerating pace.

H2 Error Rate

Our second hypothesis posits that NO DELAY marking menus have lower error rate than DELAY marking menus. The corresponding null hypothesis is that *Error Rate* differences between NO DELAY and DELAY marking menus are not significant. Figure 3 shows comparative error rates for NO DELAY in blue and DELAY in red. To test H2, we analyze *Error Rate* as a function of MENU, LAYOUT, and BLOCK. We

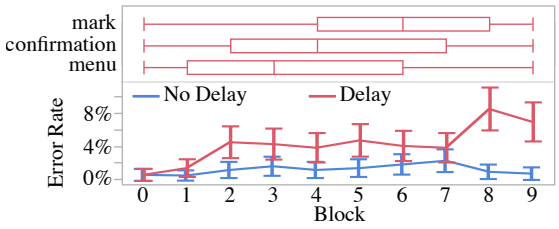

Figure 3. Top: Distribution of modes by BLOCK in the DELAY condition. Bottom: Average *Error Rates* per BLOCK and MARKING MENU condition. Error bars are 95 % CI.

found a significant effect of MARKING MENU ($F_{1,1185} = 60.88$, $p = .0001$), BLOCK ($F_{9,1185} = 4.11$, $p = .0001$), and MARKING MENU×BLOCK ($F_{9,1185} = 4.01$, $p = .0001$) on average *Error Rate*. There was no significant effect involving LAYOUT.

Post-hoc tests highlight that participants made significantly more errors with DELAY (mean 4.2 %) than with NO DELAY (mean 1.3 %) condition. This result allows us to reject the null hypothesis and claim that *H2 is supported.*

Effects of BLOCK and of the MARKING MENU×BLOCK interaction were also significant. *Error Rate* significantly *increased* with BLOCKs overall (Fig. 3-bottom), but the study of the interaction effect revealed that BLOCK levels are not significantly different from each other in the NO DELAY condition. In the DELAY condition, however, BLOCKs 9 and 8 (resp. 6.9 and 8.5 %) contain significantly more errors than BLOCKs levels 0, 1 ($<$ 1.3 %) and than every NO DELAY BLOCK ($<$ 2.6 %). This result might be explained by the fact that with more practice in this condition, participants made more selections using *mark mode* (detail analysis of *mark mode* usage below), which is more likely to result in errors because the user must know the mark corresponding to the target, compared to the *menu mode* and the DELAY condition where the menu is displayed. It also suggests that users can switch to *mark* mode before being entirely familiar with the item.

One additional question we can pose is whether the increase in error was due to decreased use of *menu mode* in the DELAY condition. Figures 2 and 3 shows the evolution of Selection time and errors, respectively. Consider, particularly, Figure 3, where we can see that error rates are identitical for DELAY and NO DELAY during BLOCK 1, but then diverge. Visually, it appears that, in DELAY, the use of *mark mode* is detrimental to accuracy. Analyzing this, we find that participants made significantly more selection errors in *mark mode* (mean 8.3 %) than in *menu* and *confirmation modes* (resp. 2 and 1.2 %).

H3 Practiced Performance

Our third hypothesis proposes that DELAY marking menus should outperform NO DELAY marking menus because mark mode should be faster to execute than menu mode, even in a NO DELAY condition because the user does not need to perceive the menu before acting. To examine H3, we need, first, an understanding of how common *mark mode* is during the DELAY. Next, we need some measure of *Practiced Use* for the NO DELAY condition, because NO DELAY has no easily distinguishable *mark mode*. Finally, we can use this understanding to contrast performance with DELAY and NO DELAY during practiced use.

To, first, explore *how common the use of mark mode is* during interaction, recall that the DELAY condition exhibits three modes, a *menu mode* (for novice use), a *mark mode* (for practiced users), and a *mark confirmation mode* to facilitate the transition between the latter two. In DELAY, we would expect use of *mark mode* to increase over time. We would also expect more complex menus to result in reduced use of *mark mode*. Graphically, Figure 3, top, shows the evolution of *mark mode* use later in the study, beginning around BLOCK 4 or 5. This observation corresponds to statistical analysis: we found a significant effect of BLOCK ($F_{9,585} = 38.5$, $p = .0001$) and LAYOUT ($F_{3,585} = 76.78$, $p = .0001$) on the use of *mark mode*. It increased overall with BLOCKs (see Fig. 3-top), ranging from 8.1 % in BLOCK 0 to 62.3 % in BLOCK 9. It also significantly decreased with menu complexity: LAYOUT 8×8 (mean 28.9 %) $\ll$ 4×4 (36.3 %) and 8 (37.2 %) $\ll$ 4 (67.4 %). There was no significant interaction effect.

Analyzing the next two questions, *some measure of Practiced Use for the* NO DELAY *condition*, and how to *contrast performance with* DELAY *and* NO DELAY *during practiced use* is complicated to disentangle. First, considering Figure 2, we see that the curve of selection time (ms) versus BLOCK for DELAY and NO DELAY mirror each other. We, therefore, first examine BLOCK effects, then look at *Execution Times* and *Preparation Times* more deeply.

Contrasting *Selection Time* across BLOCKs (Fig. 2 left) post-hoc tests showed that BLOCKs 7-8 with NO DELAY are significantly shorter (mean $<$ 1777 ms) than BLOCKs 0-5 with DELAY ($>$ 2175), but only than BLOCKs 0-2 with NO DELAY ($>$ 2139). Conversely, BLOCKs 6-9 with DELAY ($<$ 1973) were *not* significantly different from each others, nor from any NO DELAY block other than 0 (2800 ms). Given that *Selection Time* appears stable in DELAY BLOCKs 6-9, and that there is no significant difference between them and any NO DELAY blocks, we cannot, *prima facie*, reject H3's null hypothesis.

One factor that may prevent us from rejecting H3's null hypothesis is that it is hard to match a *mark mode* (DELAY) with an equivalent binary criterion in the NO DELAY condition, because the menu always appears. Perhaps participants in NO DELAY perform more "practiced" command selections.

To test this, we can assume that the "practiced" selections in any given condition form a subset of trials that span all blocks, and with distinctively good time performance. In order to properly compare practiced performance between conditions, we identify the BLOCKs in which *Selection Time* stabilized towards its minimum using Hsu's HSB contrast, i.e. blocks 7 to 9. For *Selection Time* on these "stable" BLOCKs, we found a significant effect of LAYOUT ($F_{3,105} = 60.51$, $p = .0001$), but no longer of MARKING MENU nor any interaction between the two. Discarding trials with selection errors, we found significant effects of DELAY MODE ($F_{2,414} = 331.57$, $p = .0001$) and BLOCK ($F_{9,413.1} = 2.31$, $p = .05$) on median *Selection Time*. For the DELAY condition, participants were significantly faster in *mark mode* (mean 1445 ms) than in *menu* and *confirmation*

modes (resp. 2982 and 3102 ms). However, DELAY does not outperform NO DELAY in terms of overall selection time; it remains the case that there is no significant difference between DELAY and NO DELAY during practiced use.

A final analysis we can perform involved breaking down *Selection Time* into *Execution Time* and *Preparation Time* to try to identify, with finer granularity, contrasting effects of DELAY and NO DELAY. We found a significant effect of MARKING MENU ($F_{1,105} = 7.12$, $p = .0088$) and LAYOUT ($F_{3,105} = 41.2$, $p = .0001$) on *Execution Time*, with no interaction. Post-hoc tests revealed that the DELAY condition has a smaller *Execution Time* (mean 948 vs. 1130 ms), and a similar LAYOUT effect as before. Finally, we found a significant effect of MARKING MENU ($F_{1,105} = 50.61$, $p = .0001$) and LAYOUT ($F_{3,105} = 6.32$, $p = .0007$) on *Preparation Time*, again with no interaction. Post-hoc tests revealed that the DELAY condition has a larger PREPARATION TIME (mean 830 vs. 569 ms), and a similar LAYOUT effect as before.

In summary, for practiced use, we find no significant difference between DELAY and NO DELAY in terms of *Selection Time*, and that while *Execution Time* is lower with DELAY (182 ms difference), it is compensated by a lower *Preparation Time* (261 ms difference) for NO DELAY, regardless of the LAYOUT.

STUDY 2: INVESTIGATING EXPERT PERFORMANCE

Study 1 supports both H1 and H2 – NO DELAY has significant advantages over DELAY even as users become practiced. H3, that DELAY outperforms NO DELAY in highly practiced use, was not supported; DELAY suffered from higher error rates and reaction times, even when selecting commands consistently in mark mode. However, it may be that, with sufficient practice, users could reach a theoretical level of perfect expertise – autonomic response – that would result in an overall performance benefit. Thus, study two was designed to balance the need for the "best case" of an autonomic response, versus the confound of anticipation, or the cost of deciding what selection to perform. With this design, we are able to examine the limits of expert level use to identify whether – with fully autonomic reaction – we can quantify a benefit from DELAY.

Experimental Procedure

Participants

Eight participants (average age 26.88, SD = 2.56) were recruited for the study, two who identified as female, six as male. All were either graduate students or post-doctoral researchers from technical disciplines in a technically focused university.

Task and Stimulus

The study application was a modified version of Study 1, run on the same computer and with the same mouse. Prior to completing the study, users were familiarized using the same process as study 1. The experiment followed the same procedure as study 1 except that:

- Users had to select only two items from each of two different menus repeatedly in DELAY and NO DELAY conditions.
- The two items per menu were carefully balanced through pilot studies to ensure equivalent speed and precision.

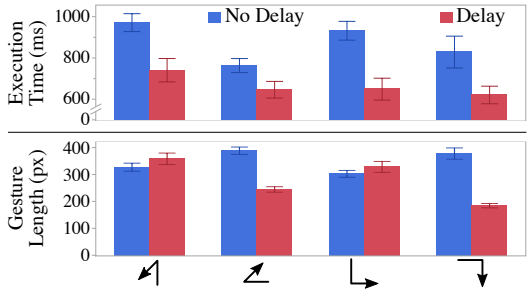

Figure 4. Effects of MENU CONDITION on *Execution time* vs *Gesture length* for each item. Mind the non-zeroed Y-axis on top.

We chose the items to select in each menu based on geometric characteristics: one acute angle and one right angle per menu. Prompted items for menu 1 were ↙ [north][south-west] and ↳ [south][east], and for menu 2 ↗ [west][north-east] and ↴ [east][south].

For each trial, the participant had to select one of the two target commands, whose label was displayed on top of the window, within a MARKING MENU of 8 × 8 (64) items. For each menu participants performed 8 BLOCKS of 10 selections per command. Ordering of the commands was randomized within each menu. Similarly to study 1, we used different command sets in each of the two 8 × 8 menus.

The result was a fully counterbalanced, within subjects, 2X2 design (MENU – DELAY/NO DELAY and ITEMS). In total, we collected 2 commands per marking menu × 10 prompts per command × 8 blocks × 2 MENUS × 8 participants = 2560 selections (640 per command) in total.

At the end of each MENU condition, we instructed participants to select the items of the corresponding pair 4 times each using arrows as instructions, to obtain a temporal floor for gesture peformance. Dependent measures and analysis method followed those of study 1.

Results

Fig 4 shows the overall *Preparation*, *Execution*, and *Selection* times throughout the study for each MENU condition. We found a significant effect of BLOCK on *Preparation time* ($F_{7,2545} = 14.59$, $p < 0.0001$), on *Execution time* ($F_{7,2545} = 49.83$, $p < 0.0001$), and on *Selection time* ($F_{7,2545} = 47.55$, $p < 0.0001$). In all three measures, BLOCK 0 took significantly longer than the rest (300 ms to 1 s), with no other significant difference between BLOCK numbers. There was no effect of BLOCK on *Error rate*. We excluded BLOCK 0 from further analyses.

We found significant effects of MARKING MENU on *Execution time* ($F_{1,2231} = 164$, $p < 0.0001$) and *Selection time* ($F_{1,2231} = 75.1$, $p < 0.0001$). Selections with DELAY were faster to *Execute* (mean 663.14 ms vs. 873.08 ms with NO DELAY), which was directly translated into *Selection time* (mean 1380.37 ms, vs. 1633.01 ms with NO DELAY), i.e. a total improvement of about 250 ms. We therefore reject H3's null hypothesis and conclude that extensive training with two targets yields faster command selection in *mark mode* with DELAY marking menus, than with NO DELAY marking menus. There was no significant effect on *Preparation time* nor on *Error rate*.

The other factor that significantly impacted time and error rate was ITEM. ITEM had a significant effect on *Execution time* ($F_{3,2229} = 16.75, p < 0.0001$) and *Selection time* ($F_{1,2229} = 6.78, p < 0.0001$). ITEM ↙ had significantly longer *Selection time* (mean 1608 ms) than ↰ and ↗ (means < 1480 ms), with ↳ in between and not significantly different than the other three. For *Execution time*, we had ↙ (mean 855 ms) ≪ ↳ (790 ms) ≪ ↰ and ↗ (means < 725 ms). We also found a significant effect of ITEM on *Error rate* ($F_{3,2229} = 2.76, p < 0.05$): ↳ (mean 2.32%) caused significantly fewer errors than ↗ (mean 5.54%). We found no effect on *Preparation time*, nor interaction effect.

Next, we explored the common hypothesis that MARK mode is faster because it allows smaller marks. We found significant effects of MARKING MENU ($F_{1,2226} = 211.48, p < 0.0001$), ITEM ($F_{3,2226} = 27.55, p < 0.0001$), and MARKING MENU × ITEM ($F_{3,28} = 5.12, p < 0.01$). DELAY caused shorter strokes overall (mean 279 vs. 349 px), and ↙ (343 px) ≫ ↗ & ↳ (316 px) ≫ ↰ (281 px). No clear pattern emerges from the Tukey test for the interaction effect, and correlations (R^2) for linear regressions between *Execution time* and *Gesture length*, for each MARKING MENU, are both 0. They all remain below 0.3 (mostly below 0.1) if regressions are performed independently for each participant. Overall, *Gesture length* seems a poor predictor of *Execution time* in the context of marking menus.

EXPERIMENT 3: ASSESSING VISUAL DISRUPTION

Our previous studies suggest that NO DELAY marking menus yield fewer errors and lower command selection times overall for all but extremely practiced targets. Kurtenbach's original design rationale for marking menus includes one additional benefit of *mark mode*, i.e. that no menu is displayed and that menus *"can be distracting"* and *"can obliterate part of the screen"* [24]. H4 presents this hypothesized benefit of DELAY. While studies 1 and 2 evaluated marking menu performance with and without delay *for prompted commands*, generalization to real-world use-cases with occlusion and visual distraction requires we test marking menus in an interactive program where occlusion of content might inhibit interaction.

To evaluate occlusion and disruption, we conducted a third experiment. In a 2 × 2 within-subjects protocol, we evaluated the effects of DELAY vs. NO DELAY marking menus in two simple yet realistic graphic arrangement applications in which participants were instructed to replicate existing figures: one involving CARTOONS and a second involving FLOW CHARTS. Our rationale for this design is twofold: it requires many repeated menu selections, so participants can exhibit practiced behaviour and possibly reach a higher level of expertise with the menus; and it creates a crowded canvas with many elements, requiring manipulation through contextual menus, so occlusion may become an issue for users.

Experimental Procedure

Participants and Apparatus

Sixteen paid participants were recruited for the study. Average age was 24.38 (SD = 2.16). Three participants identified as female and the remaining 13 identified as male. While all participants came from technical backgrounds, none of them participated in the first two studies. 2 participants had heard about Making Menus before (4 unsure, 10 no), 2 had already interacted with one (5 unsure, 9 no), and 10 had already interacted with a Pie/Radial Menu (1 unsure, 5 no).

The interface used the same display monitor and marking menu implementation as in Study 1. Mouse input was obtained through a VicTsing Slim Wireless Mouse.The experimental software was an object arrangement application in Javascript. A right button press on the page triggered a main, two-level marking menu containing the various available image items, arranged in categories, which upon selection were spawned onto the page. These images could then be dragged into and within the scene, using the left mouse button, for precise positioning. A right button press on any of those images triggered a contextual marking menu containing manipulation functions such as "rotate", "send to front", "delete", etc. whose effects were applied, upon selection, to the right-pressed image.

Task and Stimulus

As in the previous experiment, users were administered a verbal and visual demonstration of how to use marking menus within both conditions, including all modes and verbal instruction of how to use the drag and drop interface. Participants did not interact with the interface until the experimental task began. The task began with a reference image appearing in the top panel of our application (see Figure 5). Participants were then instructed to recreate the image displayed in the above panel to the below panel by "spawning" and modifying items using marking menus. Participants were asked to complete four of these tasks, two per menu condition. This experiment employed a think aloud protocol [28] inviting participants to comment while recreating the images.

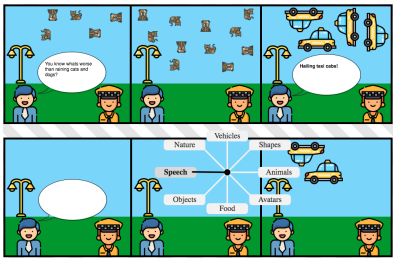

Figure 5. An example of a user interacting with the drag and drop application. On top is the figure to recreate. On the bottom is the participant's current figure with an ongoing menu selection. Icons in this figure were designed by Freepik and Smashicons from www.flaticon.com.

There were two command sets: one for CARTOONS and one for FLOW CHARTS. Each main menu had the same layout, but different configuration. For instance, the CARTOONS menu had a single one-level item in the left stroke direction and the FLOW CHARTS menu had a single one-level item in the right stroke direction. The layout consisted of a single one-level item, one five-item sub-menu category, three 4-item sub-menu categories and three 8-item sub-menu categories. The contextual menus remained consistent throughout the experiment, with a contextual menu for text (including font, style, alignment and delete) and images (including rotate or change colour, send to front, send to back and delete). Each task required the same minimum number of menu selections from each level and size of submenu.

Design and Analysis

The experiment used a 2×2 within-subjects design, with the following factors and levels:

- Main command sets / image theme: two different command sets, CARTOONS and FLOW CHARTS, were used for each menu condition to control for confounds of learning behaviour and confusion between categorical selections.
- MARKING MENU (DELAY or NO DELAY) which remained consistent with those explained in section 4.

Order of menu conditions and command sets were counterbalanced, each combination happening four times per subject.

Subjective Data Collection

Instead of determining whether occlusion occurs or if the display is "visually disruptive", our goal is to assess whether these factors affect the perceived usability of marking menus with and without delay. To compare the subjective experience of DELAY and NO DELAY, in addition to capturing spontaneous comments expressed during the tasks, participants were administered a questionnaire after each condition, comprised of Likert-scales (1 = Strongly Disagree to 7 = Strongly Agree). Questions and responses are shown in Fig. 6.

We also recorded think-aloud comments and conducted an exit interview. All subjective data (Likert, think-aloud, and interviews) were leveraged for analysis: the Likert responses as ordinal statistical data, and the qualitative data via transcripts and focused coding regarding occlusion and disruption.

Results

Practiced performance

When users can decide the order they wish to perform the overall task it, becomes a challenge to assess selection errors. To confirm that the study was long enough for users to become experienced with the menu layouts, we counted selections in the DELAY condition performed in each available mode (menu, mark, confirmation) that were not cancelled before completion. Among the 1279 successful command selections performed by all 16 participants in DELAY condition, 48% (615, mean 38.4 per participant, SD 17.5) were performed entirely in *menu* mode, 18% (225, mean 15/p, SD 9.2) involved in *confirmation* mode, where participants pause upon completion to verify whether they successfully selected the command, and 34% (439, mean 31.4/p, SD 19.7) were performed entirely in *mark* mode. Because over 1/3 of DELAY commands were invoked in *mark mode*, we believe that this indicates that participants were, generally, able to learn the menus.

Subjective preferences

Some overall trends appeared for both conditions. Questions on performance with items whose location is already remembered (a, b) received neutral or positive scores (4-7). Participants were not generally bothered by occlusion: only one scored above 5 to question (g) (NO DELAY condition). Finally, only two participants scored below neutral (4) to memorization (e) and real-world use (i) questions, both for DELAY.

An ordinal logistic fit found a single significant effect of MARKING MENU on the answers to question (h): "The [DELAY or ABSENCE OF DELAY] made me lose focus on my main task" ($\chi^2 = 3.97$, $p = .046$), with DELAY (mean score 4, most frequent scores 1, 3, 4, and 7 each with N=3) found to be more problematic than ABSENCE OF DELAY (mean 2.56, most frequent score 1 with N=7). For all other questions, we found no significant effect of MARKING MENU. Ratings are summarized in Fig. 6.

At the end of the study, 7 participants preferred DELAY, 7 preferred NO DELAY, and 2 expressed no preference.

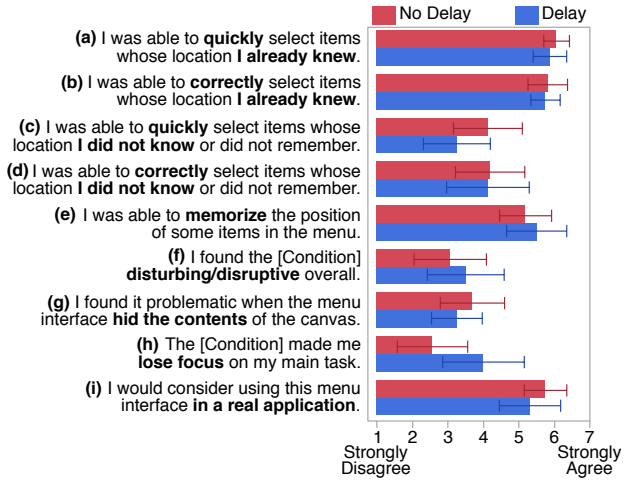

Figure 6. Likert scale questions (error bars are 95% CI).

Subjective Comments

We did not observe a consensus among participants against DELAY or NO DELAY marking menu in terms of disruption or disturbance. For example, P_8 said they *"didn't find any disturbance [in (NO DELAY)]"* and P_{12} mentioned *"I noticed no disturbance or disruption [in NO DELAY] because I could always control visibility of the marking menu by simply releasing the right mouse button"*.

A single participant (P_6) reported an issue with object occlusion in the NO DELAY condition, specifically when attempting to rotate an arrow image: *"I cannot see which direction the arrow currently is, so I don't know how to rotate it. The arrow is hidden, the menu should come somewhere else without hiding the picture [...] The menu should not hide the existing element about to be manipulated"*. That being said, this particular issue impacted only one participant and would be present in either condition to a user unfamiliar with the menu layout. One can expect that a user familiar with the menu would not forget the orientation of the arrow while performing the command selection gesture. In fact, P_{12} noted, in the DELAY condition, *"the fact that the menu hid some of the content was only problematic because I knew that closing the menu just to see what's behind would come at the additional cost of waiting when re-opening it later"*.

In terms of speed, echoing findings in the initial experiment and Likert questions (a) and (c), P_{15} noted, *"I've been working with Photoshop, it's basically like this [system] ... the user wants to do the process as fast they can, so they don't want to wait [in the DELAY condition], it's slow"*.

DISCUSSION AND CONCLUSION

Delay has been an integral component of menu mode activation in marking menus since their introduction. Despite this, little has been done to explicitly evaluate the costs and benefits of marking menu delay to learning, performance, and visual disruption. There is good reason to evaluate this trade-off, however, because there exists a theoretical tension around the use of delay. Specifically:

- Delay penalizes the novice experience, which may frustrate learners – but may also promote faster learning.
- Delay stresses recall over recognition, which may lower throughput via errors and high cognitive load – but may also speed performance for well-known, frequent commands.
- Delay eliminates visual occlusion of content and visual disruption of the interface for practiced commands – but it is unclear how costly visual occlusion and disruption are during command invocation, especially if it is expected.

Our hypotheses probe these questions, exploring overall cost during learning and into early practiced use [H1, Experiment 1], error rate [H2, Experiment 1], speed during highly practiced use [H3, Experiments 1 and 2], and perceived visual disruption [H4, Experiment 3]. We find support for H1 and H2, that DELAY does significantly impact the cost of marking menus. We also find mixed results for H3, with no benefit for DELAY observed during *mark mode* activations in experiment 1 and only a benefit during autonomic command invocation in experiment 2. Finally, in experiment 3, we note limited issues around visual occlusion despite the fact that *mark mode* was used 34% of the time during the DELAY condition; in fact, in experiment 3, we noted significantly more impact due to the cost of DELAY than issues with occlusion.

Our difficulty finding support for H3 and H4 was somewhat surprising, as the assumption with marking menu is that *mark mode* use should have performance benefits, and that *mark mode* eliminates visual disruption. Temporal benefits of *mark mode* were difficult to identify in experiment 1. Visual disruption issues were difficult to identify in experiment 3. Only in study 2 did we identify a statistically significant benefit of *mark mode*, and then only for two highly practiced commands.

The results of study 1 and 2 in concert suggest that extensive training may be required to overcome performance costs, and it remains unclear whether users would be able to reach expertise with enough commands for the delay to be beneficial overall when novice, practiced, and expert behaviours are considered altogether. Assuredly, it is virtually impossible for a user to remember a complete menu layout: even projections using a Zipfian distribution in a 8 items only marking menu suggest that users would be far from selecting all commands in *mark mode* after extensive use, selecting only 43% when delay is 200 ms delay, 55% with 333ms, and only reaching a higher rate with significantly longer delays, up to 2s, but at the cost of a doubling or tripling of error rate [27].

The most controversial result in our work is the lack of subjective impact of visual disruption in Experiment 3. However, in hindsight it is questionable whether this result should be surprising: consider past work by Cockburn et al. on menu performance [7, 8] and past work on "mousing ahead" by Hopkins [18, 20]. In Cockburn et al.'s work, novice (or unpracticed) performance is modeled via linear visual search because the user still needs to read each menu item to find their desired target, and expert (practiced) performance is modeled via the Hick-Hyman Law because the user simply mentally selects the desired action from among candidates without the need to look for menu options. Cockburn et al.'s model includes no temporal cost that results from visual disruption in their model of menu access, and their model correlates perfectly with user performance. Regardless, visual disruption may also be an over-stated issue: while one way to eliminate visual disruption caused by occlusion is to avoid displaying the menu, partial transparency of the menu may also allow users to continue to see underlying content without the need to include novice mode penalty and increased error rate in marking menus.

While our results begin to explore the costs and benefits of DELAY in marking menu interfaces, additional work remains. In our email questionnaire, Kurtenbach noted that, in industry, he has used delay values significantly shorter than those in the academic literature – around 100ms, and that many users fail to discover *mark mode*. These shorter delays may reduce error, speed novice performance, and still support autonomic performance as per experiment 2. We also note limitations in our work, including a relatively small sample size and the lack of an in-situ scenario, i.e., utilizing an application where marking menus are actually used, such as Audodesk's Maya.

Our results open similar questions regarding other command selection techniques that rely on delay-separated modes [11, 12, 14, 16, 17] to implement the rehearsal design principle. Future work should investigate whether similar results would be found with these interfaces. For example, the FastTap technique displays a full-screen grid that may result in higher perceived visual disturbance.

In the end, our results are valuable to designers exploring the use of Marking Menus and other rehearsal-based interfaces. Understanding DELAY's benefit for highly practiced commands allows designers to choose to incorporate delay, depending on their perspective of whether the system being designed should penalize novice and casual use in order to optimize highly practiced expert use. This paper is, to our knowledge, the first quantitative and qualitative exploration of the relative costs of these factors in rehearsal-based interface design.

ACKNOWLEDGEMENTS

We thank participants in our study. Experiments in this work were approved by the Office of Research Ethics at the University of Waterloo. This research was supported by the Natural Science and Engineering Research Council of Canada (NSERC), the National Research Council of Canada (NRC) via the NRC-University of Waterloo Collaboration Centre (NUCC), and the LAI Réapp.

REFERENCES

[1] Gilles Bailly, Eric Lecolinet, and Yves Guiard. 2010. Finger-count and Radial-stroke Shortcuts: 2 Techniques for Augmenting Linear Menus on Multi-touch Surfaces.

In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 591–594. DOI: http://dx.doi.org/10.1145/1753326.1753414

[2] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2007. Wave Menus: Improving the Novice Mode of Hierarchical Marking Menus. In *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction (INTERACT'07)*. Springer-Verlag, Berlin, Heidelberg, 475–488. http://dl.acm.org/citation.cfm?id=1776994.1777053

[3] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2008. Flower Menus: A New Type of Marking Menu with Large Menu Breadth, Within Groups and Efficient Expert Mode Memorization. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '08)*. ACM, New York, NY, USA, 15–22. DOI: http://dx.doi.org/10.1145/1385569.1385575

[4] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2016. Visual Menu Techniques. *ACM Comput. Surv.* 49, 4, Article 60 (Dec. 2016), 41 pages. DOI: http://dx.doi.org/10.1145/3002171

[5] Olivier Bau and Wendy E. Mackay. 2008. OctoPocus: A Dynamic Guide for Learning Gesture-based Command Sets. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, New York, NY, USA, 37–46. DOI: http://dx.doi.org/10.1145/1449715.1449724

[6] Mathieu Berthellemy, Elodie Cayez, Marwan Ajem, Gilles Bailly, Sylvain Malacria, and Eric Lecolinet. 2015. SpotPad, LociPad, ChordPad and InOutPad: Investigating Gesture-based Input on Touchpad. In *Proceedings of the 27th Conference on L'Interaction Homme-Machine (IHM '15)*. ACM, New York, NY, USA, Article 4, 8 pages. DOI: http://dx.doi.org/10.1145/2820619.2820623

[7] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2014. Supporting Novice to Expert Transitions in User Interfaces. *ACM Comput. Surv.* 47, 2, Article 31 (Nov. 2014), 36 pages. DOI: http://dx.doi.org/10.1145/2659796

[8] Andy Cockburn, Per Ola Kristensson, Jason Alexander, and Shumin Zhai. 2007. Hard Lessons: Effort-inducing Interfaces Benefit Spatial Learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 1571–1580. DOI: http://dx.doi.org/10.1145/1240624.1240863

[9] Fergus IM Craik and Robert S Lockhart. 1972. Levels of processing: A framework for memory research. *Journal of verbal learning and verbal behavior* 11, 6 (1972), 671–684.

[10] Brian D. Ehret. 2002. Learning Where to Look: Location Learning in Graphical User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 211–218. DOI: http://dx.doi.org/10.1145/503376.503414

[11] Bruno Fruchard, Eric Lecolinet, and Olivier Chapuis. 2017. MarkPad: Augmenting Touchpads for Command Selection. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 5630–5642. DOI: http://dx.doi.org/10.1145/3025453.3025486

[12] Varun Gaur, Md. Sami Uddin, and Carl Gutwin. 2018. Multiplexing Spatial Memory: Increasing the Capacity of FastTap Menus with Multiple Tabs. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18)*. ACM, New York, NY, USA, Article 22, 13 pages. DOI: http://dx.doi.org/10.1145/3229434.3229482

[13] Emmanouil Giannisakis, Gilles Bailly, Sylvain Malacria, and Fanny Chevalier. 2017. IconHK: Using Toolbar Button Icons to Communicate Keyboard Shortcuts. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4715–4726. DOI: http://dx.doi.org/10.1145/3025453.3025595

[14] Alix Goguey, Sylvain Malacria, Andy Cockburn, and Carl Gutwin. 2019. Reducing Error Aversion to Support Novice-to-Expert Transitions with FastTap. In *Actes de la 31e conférence francophone sur l'Interaction Homme-Machine (IHM 2019)*. ACM, Grenoble, France, 1:1–10. DOI: http://dx.doi.org/10.1145/3366550.3372247

[15] Tovi Grossman, Pierre Dragicevic, and Ravin Balakrishnan. 2007. Strategies for Accelerating On-line Learning of Hotkeys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 1591–1600. DOI:http://dx.doi.org/10.1145/1240624.1240865

[16] Carl Gutwin, Andy Cockburn, and Benjamin Lafreniere. 2015. Testing the Rehearsal Hypothesis with Two FastTap Interfaces. In *Proceedings of the 41st Graphics Interface Conference (GI '15)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 223–231. http://dl.acm.org/citation.cfm?id=2788890.2788930

[17] Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. 2014. Faster Command Selection on Tablets with FastTap. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2617–2626. DOI: http://dx.doi.org/10.1145/2556288.2557136

[18] Don Hopkins. 1991. The Design and Implementation of Pie Menus. *Dr. Dobb's J.* 16, 12 (Dec. 1991), 16–26. http://dl.acm.org/citation.cfm?id=135025.135026

[19] Per-Ola Kristensson and Shumin Zhai. 2004. SHARK2: A Large Vocabulary Shorthand Writing System for Pen-based Computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 43–52. DOI:http://dx.doi.org/10.1145/1029632.1029640

[20] Gordon Kurtenbach and William Buxton. 1991. Issues in Combining Marking and Direct Manipulation Techniques. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology (UIST '91)*. ACM, New York, NY, USA, 137–144. DOI: http://dx.doi.org/10.1145/120782.120797

[21] Gordon Kurtenbach and William Buxton. 1993. The Limits of Expert Performance Using Hierarchic Marking Menus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93)*. ACM, New York, NY, USA, 482–487. DOI: http://dx.doi.org/10.1145/169059.169426

[22] Gordon Kurtenbach and William Buxton. 1994. User Learning and Performance with Marking Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*. ACM, New York, NY, USA, 258–264. DOI: http://dx.doi.org/10.1145/191666.191759

[23] Gordon Kurtenbach, George W. Fitzmaurice, Russell N. Owen, and Thomas Baudel. 1999. The Hotbox: Efficient Access to a Large Number of Menu-items. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 231–237. DOI: http://dx.doi.org/10.1145/302979.303047

[24] Gordon Paul Kurtenbach. 1993. *The design and evaluation of marking menus.* University of Toronto Toronto.

[25] Benjamin Lafreniere, Carl Gutwin, Andy Cockburn, and Tovi Grossman. 2016. Faster Command Selection on Touchscreen Watches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4663–4674. DOI:http://dx.doi.org/10.1145/2858036.2858166

[26] Kris M. Y. Law, Victor C. S. Lee, and Y. T. Yu. 2010. Learning Motivation in E-Learning Facilitated Computer Programming Courses. *Comput. Educ.* 55, 1 (Aug. 2010), 218–228. DOI: http://dx.doi.org/10.1016/j.compedu.2010.01.007

[27] Blaine Lewis. 2019. *Longer Delays in Rehearsal-based Interfaces Increase Expert Use*. Master's thesis. University of Waterloo.

[28] Clayton Lewis and John Rieman. 1993. Task-centered user interface design. *A Practical Introduction* (1993).

[29] Frank Linton, Deborah Joy, and Hans-Peter Schaefer. 1999. Building User and Expert Models by Long-Term Observation of Application Usage. In *UM99 User Modeling*, Judy Kay (Ed.). Springer Vienna, Vienna, 129–138.

[30] Yuexing Luo and Daniel Vogel. 2015. Pin-and-Cross: A Unimanual Multitouch Technique Combining Static Touches with Crossing Selection. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. ACM, New York, NY, USA, 323–332. DOI: http://dx.doi.org/10.1145/2807442.2807444

[31] Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. 2013. Promoting Hotkey Use Through Rehearsal with ExposeHK. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 573–582. DOI: http://dx.doi.org/10.1145/2470654.2470735

[32] Mathieu Nancel, Stéphane Huot, and Michel Beaudouin-Lafon. 2009. Un Espace De Conception Fondé Sur Une Analyse Morphologique Des Techniques De Menus. In *Proceedings of the 21st International Conference on Association Francophone D'Interaction Homme-Machine (IHM '09)*. ACM, New York, NY, USA, 13–22. DOI: http://dx.doi.org/10.1145/1629826.1629829

[33] Stuart Pook, Eric Lecolinet, Guy Vaysseix, and Emmanuel Barillot. 2000. Control Menus: Excecution and Control in a Single Interactor. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems (CHI EA '00)*. ACM, New York, NY, USA, 263–264. DOI:http://dx.doi.org/10.1145/633292.633446

[34] Anne Roudaut, Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2009. Leaf Menus: Linear Menus with Stroke Shortcuts for Small Handheld Devices. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part I (INTERACT '09)*. Springer-Verlag, Berlin, Heidelberg, 616–619. DOI: http://dx.doi.org/10.1007/978-3-642-03655-2_69

[35] Quentin Roy. 2018. Marking Menu implementation in JavaScript. https://github.com/QuentinRoy/Marking-Menu. (2018).

[36] Quentin Roy, Sylvain Malacria, Yves Guiard, Eric Lecolinet, and James Eagan. 2013. Augmented Letters: Mnemonic Gesture-based Shortcuts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2325–2328. DOI: http://dx.doi.org/10.1145/2470654.2481321

[37] Joey Scarr, Andy Cockburn, and Carl Gutwin. 2013. Supporting and Exploiting Spatial Memory in User Interfaces. *Found. Trends Hum.-Comput. Interact.* 6, 1 (Dec. 2013), 1–84. DOI: http://dx.doi.org/10.1561/1100000046

[38] Shengdong Zhao, Maneesh Agrawala, and Ken Hinckley. 2006. Zone and Polygon Menus: Using Relative Position to Increase the Breadth of Multi-stroke Marking Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 1077–1086. DOI:http://dx.doi.org/10.1145/1124772.1124933

[39] Shengdong Zhao and Ravin Balakrishnan. 2004. Simple vs. Compound Mark Hierarchical Marking Menus. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 33–42. DOI: http://dx.doi.org/10.1145/1029632.1029639

[40] Jingjie Zheng, Xiaojun Bi, Kun Li, Yang Li, and Shumin Zhai. 2018. M3 Gesture Menu: Design and Experimental Analyses of Marking Menus for Touchscreen Mobile Interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 249, 14 pages. DOI: http://dx.doi.org/10.1145/3173574.3173823