Longer Delays in Rehearsal-based Interfaces Increase Expert Use

by

Blaine Lewis

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2019

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis includes first-authored material currently under peer-review in a journal submission published by the Association for Computing Machinery (ACM). The manuscript from which I have adapted content is the following:

- Blaine Lewis and Daniel Vogel. Longer Delays in Rehearsal-based Interfaces Increase Expert Use. *ACM Transactions on Computer-Human Interaction (TOCHI)*. 36 pages. New York, NY, USA. ACM.

Abstract

Rehearsal-based interfaces are designed to encourage a transition from novice to expert, but many users fail to make this transition. Most of these interfaces activate novice mode after a short delay, between 150 and 500ms. Our work investigates the impact of this delay time on expert usage and learning in three crowdsourced experiments. The first experiment examines an 8-item marking menu with delay times ranging from 200ms to 2 seconds. Results show longer delays increase successful expert selections. The second and third experiments generalise this result to a different rehearsal-based menu, a desktop clone of FastTap with 8-items and 15-items. Together, our results show that expert use correlates with delay time, but delay time does not always improve menu memorisation. However, imperceptible delays of 200ms harm long term retention of menu items. Designers of rehearsal-based interfaces should take advantage of longer delays to encourage a transition to expert usage.

Acknowledgements

I would like to thank my advisor, Dan Vogel, who's enormous trust in my independence has driven me to reach for things I thought were out of my reach. His continual support and understanding of how I work best has let me not only succeed in my master's but also enjoy the process.

The entire HCI lab has driven me to be a better person and researcher than I would have thought possible before coming here. Greg for one wouldn't tolerate me to be less than a good person and I have the utmost respect for him because of that. I'll miss being able to pitch my crazy ideas in person to anybody who will listen. The people were my favourite part of my degree and I hope we never spread too far apart.

Finally, I'd like to thank my friends and family back home for being there when I'm back even if it's never often enough.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Many user interfaces provide novice and expert interaction techniques. For example, in a typical graphical user interface, novices trigger a command using a menu, but experts trigger the same command using a keyboard short-cut. Expert techniques like this are more efficient, but many novices do not transition to using them [37]. One reason is when there is a disconnect between novice and expert techniques. in the example above, selecting a menu item with a mouse is very different than pressing a shortcut key [29].

Rehearsal-based interfaces are designed to smooth the transition from novice to expert by providing a single interaction technique with an optional visual guidance mode for novice users. This novice guidance is often activated after the user delays their input action for some time threshold. The thought is that this delay prevents novice guidance from distracting expert users [30] and improves learning through effort [13]. In practice, rehearsal-based interfaces have employed a range of delay times. Kurtenbach and Buxton used a 333ms delay when they originally proposed such a “press-and-wait” novice mode activation for their marking menu [29]. Afterwards, other rehearsal-based techniques were proposed using various delay times, such as 150ms [41], 200ms [23], 250ms [35], 300ms [54], 400ms [53], and 500ms [18]. Some delays are not reported [39, 56], or only described as a “fraction-of-a-second” [3].

The question is, how critical is a chosen delay time? Does it have any measurable impact on expert usage, learning retention, time, errors, or user frustration? We introduce a descriptive model of rehearsal-based interfaces that emphasises a user’s cognitive decision to choose between novice or expert mode and the key role of a *reactivator* (the rehearsal mode activation method) with which that user exercises their decision to use novice mode. This simple model provides a way to examine possible factors influencing a user’s decision and how the design of the reactivator might nudge them towards using expert mode. A delay is arguably the simplest form of reactivator, and longer delays increase the effort to use it. A relationship between cognitive effort and learning has been shown in psychology [52] and Cockburn et al. showed that high input effort increased expert use [13]. So it stands to reason that increased delay time could result in higher rates of expert mode adoption and learning for rehearsal-based interfaces.

We test this hypothesis in three online crowdsourced experiments designed to measure the rate of expert mode selections, selection errors, level of learning and retention using recall tests, as well as subjective measures like frustration. The first experiment uses a classic 8-item marking menu with delays spanning an almost imperceptible, but controllable 200ms, to a very noticeable 2 seconds. Our results show an increase in successful expert selections by up to 39% for longer delays, with some increase in error rates but no detectable difference in frustration. Although longer delays do not necessarily increase the level of learning (measured by item location retention), we found evidence that very short delays can harm retention. The second experiment generalises these results using a different 8-item rehearsal-based menu, a desktop version of FastTap [24] that we call Fast2Click, and the third experiment further demonstrates how our results extend to a larger 15-item Fast2Click menu.

This thesis contributes empirical evidence that rehearsal-based interfaces should use longer delays. We believe the reason may be that longer delays make novice mode feel like a penalty, so users become willing to try and use expert mode, even if they make more errors initially. This effect extends to different rehearsal-based menus, and menus with more items. In sum, we contribute new results for understanding and designing rehearsal-based interfaces.

1.1 Contributions

This thesis makes the following contributions:

- A descriptive model of a rehearsal-based interface into components that connect a user's transition from novice to expert.
- Formalising the concept of a reactivator, the delimiter separating a novice mode selection from an expert mode selection.
- A set of three crowdsourced experiments: the first exploring the effect of delay on user performance for traditional marking menus; a second experiment generalising

those effects to other rehearsal-based interfaces; and a final experiment using larger command sets.

- Design implications for both academic literature and interfaces found in real applications.

1.2 Organisation

- Chapter 2 provides background information on the transition from novice to expert and a review of rehearsal-based interfaces. It concludes by describing a descriptive model of rehearsal-based interfaces.
- Chapter 3 describes the first crowdsourced experiment investigating the effect of delay on users of marking menus.
- Chapter 4 extends the experiment described in Chapter 3 and explores if the effects generalise to another form of rehearsal-based interface.
- Chapter 5 presents a final experiment with more menu items to see if the effects seen in the previous two experiments extend to menus with more items.
- Chapter 6 explains the implications of the findings from the experiments described in Chapters 3-5.

Chapter 2

Background and Related Work

In this chapter we define what a transition from novice to expert interaction is and possible barriers impeding it. We review methods for overcoming those barriers, with special attention to rehearsal-based interfaces and the role of a delay in its various implementations. Based on prior work, we argue delay length is an important but overlooked factor for rehearsal-based interfaces, and we motivate the need for a model to define its role and a formal evaluation to measure its effect.

2.1 Novice to Expert Transition

In general, Cockburn et al. [12] define a novice to expert transition to be when a user reaches a "high level of performance" while completing a task with a computer interface. Performance typically means rapid completion time, but it could also mean fewer errors, a higher quality result, or similar measures. Along with Cockburn et al., Scarr et al. [47] suggest there are two forms of improvement that lead to novice to expert transitions: *intramodal improvement* is within a single method of interaction and *intermodal improvement* is between different interaction methods.

An *intermodal improvement* results from switching to an interaction method with a higher performance ceiling, and using that new method effectively [47]. A common example of an intermodal improvement is switching from selecting a command from a linear menu to triggering the same command with a keyboard shortcut. It may seem redundant to have two interaction techniques to perform the same command, but different interaction techniques support different needs. For example, keyboard shortcuts allow for highly efficient interaction, whereas linear menus support learning and discovery of commands [37]. Despite the benefits of transitioning from interaction techniques designed for discoverability to those designed for efficiency, users often fail to make the transition.

Carroll and Rossen suggest that users fail to transition to expert techniques due to a production bias and an assimilation bias [9]. A production bias pushes users to focus on the short term goal of maximising immediate throughput rather than the long term goal of maximising throughput in the future. Shortsighted goals mean that users will not spend time (and reduce current throughput) in order to improve throughput in the future. Scarr et al. quantified production bias using two learning curves, and postulated that users fear experiencing a performance dip when first remembering and learning the expert method [47]. Users also typically have an assimilation bias where they apply old knowledge to new problems rather than finding a more efficient technique. Blur [47] uses a calm notification strategy to inform users of the existence of a more efficient technique to select a command. By notifying users of more efficient techniques they can help solve the assimilation bias, and help users become more aware of the production bias.

Another barrier to intermodal transitions is “satisficing.” Fu and Gray suggest that users *satisfice* [19], meaning they become so satisfied with a “good enough” method that they do not seek new methods to further improve performance. After becoming aware of a new method, a user’s perception of the new technique must be in favour of switching, and by a large margin to avoid satisficing [19, 50]. Satisficing is an even greater issue given that users often incorrectly perceive the performance difference between two methods. For example, many participants of Tak et al.’s study mistakenly thought that toolbar-based command selections were faster than keyboard shortcuts. However, extensive research [37, 42] shows that keyboard shortcuts are much faster than toolbars and their use could potentially save days of productivity over the long term. Skillometers [43] provide a subtle visualisation of performance improvements possible by performing an intermodal switch. By increasing awareness of potential gains, users are more likely to switch to more efficient interaction techniques.

Intramodal improvement occurs when a user improves their performance when using a single interaction technique. For example, executing a keyboard shortcut repeatedly will reinforce a user’s memory of the keyboard shortcut and improve the motor action required to press the key. Intramodal learning has three stages that align with Fitts’ and Posner’s stages of motor learning [17]: initial performance, extended learning, and ultimate performance.

Cockburn et al. suggest that initial performance can be improved by making visual search better, such as Ephemeral Adaptation [16] which reduces the number of items displayed. They also suggest three design considerations for improving extended learning: "effortful learning" where artificial barriers to performance increase the rate of learning [13]; Kurtenbach's principles of revelation, guidance and rehearsal [31]; and avoiding "overguidance" where users become too reliant on guidance as demonstrated by the guidance hypothesis in motor learning [48]. Scarr et al. suggest that the ultimate performance of an interaction technique can be improved by using flat command structures, terse and expressive input, history support, and spatial predictability [47]. Intramodal improvements are usually less drastic than intermodal improvements, but a user who achieves automaticity [17] does not need to waste effort thinking about command selection, and they can instead focus on the main task.

An important aspect to consider for learning is how to measure it. Anderson and Bischof [1] argue, and demonstrate in an experiment, that measuring recall immediately after a training task only captures short-term learning. They show it is more critical to measure recall after 24 hours to understand longer term learning retention.

2.2 Rehearsal-based Interfaces

We follow Gutwin et al.'s definition [23] of a rehearsal-based interface to mean an interface that embodies Kurtenbach's principle of rehearsal: "Guidance should be a physical rehearsal of the way an expert would issue the command" [31]. The best way to explain the concept is by describing specific interaction mechanics used in two popular examples of rehearsal-based interfaces.

Kurtenbach's original mouse-controlled marking menu [32, 29, 30] is the classic example of a rehearsal-based interface. Selecting a command using expert mode is accomplished by simply pressing the mouse button, drawing a "mark" (a line representing the input motion), and releasing the mouse button. Novice users can initiate a guided mode by pressing the mouse button down and then waiting for a "press-and-wait" delay, after which, visual guidance in the form of command names presented at radial locations is displayed. Using this guidance, the novice user draws a mark towards the desired command in the same way an expert does, rehearsing the expert interaction technique. A study examining marking menus suggests marking menus are faster than pie, radial, and linear menus [30].

FastTap [24, 23] is another rehearsal-based interface designed for multitouch input on a tablet. An activation button is anchored at the bottom-corner of the screen. A novice user places their thumb on the button, and after a short delay, a grid of commands is displayed for guidance. While keeping their thumb on the activation button, they use another finger of the same hand to press the desired command in the grid. Experts can perform the same thumb and finger interaction without waiting for the grid of commands to appear. A study shows that FastTap outperforms hierarchical marking menus on a tablet [24].

The features of rehearsal-based interfaces allow for a smoother transition from novice to expert. Rehearsal allows a novice to practice an expert interaction using visual guidance. Intramodal improvements like more confident and faster mark movement using novice mode should transfer directly to expert mode as well. Expert mode allows for a higher performance ceiling since there are no distracting visuals which could lead to overguidance [48], and may allow users to transition from a visual search task to memory retrieval resulting in faster selections. Finally, the interaction similarities between novice and expert mode reduce the chance of a performance dip when users make an intermodal transition.

Despite their strength for helping users transition to expertise, two studies using FastTap show that rehearsal-based interfaces do not always result in users switching longterm to the expert technique. Gutwin et al. demonstrate that intermodal switches do occur when users perform tasks with a high level of urgency in a game-like countdown time setting [23]. But when tasks have less urgency, such as selecting tools and colours in a painting application, expert mode usage is not encouraged in the same way. Lafreniere et al. study the persistence of expert adoption after training using FastTap for different applications [35]. In practice, users exhibit an “all-or-nothing” effect where some use the expert mode for nearly all selections, and others for almost no selections. The results from Lafreniere et al. and Gutwin et al. suggest that rehearsal-based interfaces could be better designed to encourage an intermodal switch from novice use to expert use.

2.3 Delays Used in Rehearsal-based Interfaces

A key part to rehearsal-based interfaces is how the user chooses between novice and expert mode. Like the Marking Menu and FastTap examples above, this is most often accomplished by requiring the user to pause for a short time before guidance is shown. This is convenient because it naturally prevents overguidance, but Kurtenbach and Buxton also argue delayed guidance prevents user distraction, reduces screen occlusion, and limits system processor demand [31].

So delay serves an important function in rehearsal-based interfaces, but the specific delay times used have varied greatly. Kurtenbach and Buxton [28] used a 333ms delay for marking menus. The Flower Menu [4] increased the expressive input space of a marking menu, and followed suit with a 333ms delay. The Wave Menu [3] improves novice mode by creating better visualisations, but does not indicate the delay time other than a "fraction of a second." The Blender 3D modelling application[1] has pie menus which have a default 100ms second delay. Finally, Lepinski et al. expanded marking menus to multitouch [39], but the delay time is not mentioned.

Techniques that extend the principle of rehearsal-based interfaces beyond marking menus also use a delay. Octopocus [6] provides a feedforward guide to help users learn mouse gestures after pausing for 250ms. This delay and guidance can be triggered in the middle of an interaction, similar to the marking menu intermediate mode. Many touch-based interfaces tend to use shorter delays, for example Pin-and-Cross [41] uses a 150ms delay for a multitouch crossing radial menu. FastTap [24, 23, 35] uses 200 to 250ms for a "two-tap" menu using a spatially stable grid of commands spanning a tablet screen. Lafreniere et al. extended FastTap [36] to a smartwatch, also using a delay of 250ms. Handmark [54] uses a 300ms delay for a multitouch spatially stable menu surrounding the user's other hand. MarkPad [18] repurposes a laptop trackpad for command selection using tactile marks to create a rehearsal-based menu with both a novice mode and "semi-novice mode". Novice users begin a selection using a function button to show a full menu and semi-novice users delay for 500ms to show part of the menu. Uddin et al. applied increasingly longer delays with respect to the complexity of spatially-stable grid menus (called CommandMaps) [53]: 200ms for 64 items, 350ms for 96 items, and 400ms for 160 items. They conjectured that longer delays might aid memorisation with more items.

In previous work, designers of rehearsal-based interfaces have chosen delays from 150 to 500ms with little rational to suggest how the delay was chosen.

2.4 Delays to Improve Learning and Other Performance

Although the original delay was created partially to prevent expert users from being distracted by the display of a menu [30], it forces users to expend effort when making a selection. Results from psychology indicate that expending cognitive effort has a positive effect on memory [52]. These effects have been applied to Human-Computer Interaction, where a harder-to-use "frost-brushing" interface [13] yielded better performance after practice with no measurable increase in frustration. Cockburn et al. suggest that effortful learning improves the extended learning of an interaction technique [12]. A delay should have a

[1] https://docs.blender.org/manual/en/dev/preferences/interface.html#menus

similar effect by giving users an opportunity to consider the menu locations while they wait for the delay to pass. Perhaps a delay up to a second longer than the original 333ms could help users remember menu locations?

Delays have also been used in non-rehearsal-based interfaces to address aspects of performance. Ephemeral adaptation [16] reduces cognitive load by delaying the display of some menu items in a linear menu. They found that a longer delay of 500ms improves selection speed. Grossman et al. suggested several methods to improve learning keyboard shortcuts, including a two second delay that acts as a penalty when users make selections using menus rather than shortcut keys [21]. Lockout delays prevent users from interacting with a system, such delays of 10 seconds have been shown by Back et al. [2] to reduce errors when completing routine data entry for critical tasks. Although rehearsal-based interfaces have only employed delays less than 500ms, given that longer delays have been used in other interfaces, a longer delay in rehearsal-based interfaces might improve performance without increasing frustration.

In summary, the literature shows that rehearsal-based interfaces have the potential to address many of the factors hindering a transition to expert techniques. However, rehearsal-based interface implementations have used many different delays without empirical evidence supporting how long the delay should be. Literature from the Cognitive Sciences and Human-Computer Interaction indicate that effort plays a significant role in the performance of an interface, and it seems likely that a delay could stimulate effort. Before we describe our experiments to test this idea, we first construct a descriptive model of rehearsal-based interfaces to help analyse the impact that delay has on the transition from novice to expert.

Chapter 3

A Descriptive Model of Rehearsal-based Interfaces

Carroll defines a descriptive model as “a framework or context for thinking about or describing a problem or situation” ([8] p30). For example, Buxton’s Three-state Model of Graphical Input [7] is a classic example of a descriptive model used to discuss issues and challenges related to input devices. Our goal is similar, we wish to define terminology, states, and components for command selection with rehearsal-based interfaces. In particular, our model enables us to situate the role of delay time within the context of novice and expert mode interactions.

The model consists of four components, an activator, an interaction, a reactivator, and a guide (Figure 3.1), and it highlights the role of the user’s decision to use expert mode or novice mode. The *activator* delimits a command selection from previous interactions and notifies the system that a rehearsal-based interface is activate, and a command selection will come next. In a marking menu, the initial mouse-down event is the activator. After activating, the user must execute their *decision* to use novice or expert mode. An expert simply performs an *interaction* which results in a command selection. With a marking menu, the interaction is drawing a directional mark. If, however, the user decides to use novice mode, they must first use a *reactivator* to explicitly activate the novice mode. The term reactivator is short for “rehearsal mode activator” since the novice mode is a way to rehearse the expert action. In a marking menu, the reactivator is the “press-and-wait” delay time. After the reactivator is performed, a *guide* interface is presented to indicate to a novice user how to perform an interaction to make a specific command selection. In a marking menu, the guide is a set of menu item names displayed in a circular array indicating which direction to draw a line mark to activate the corresponding command.

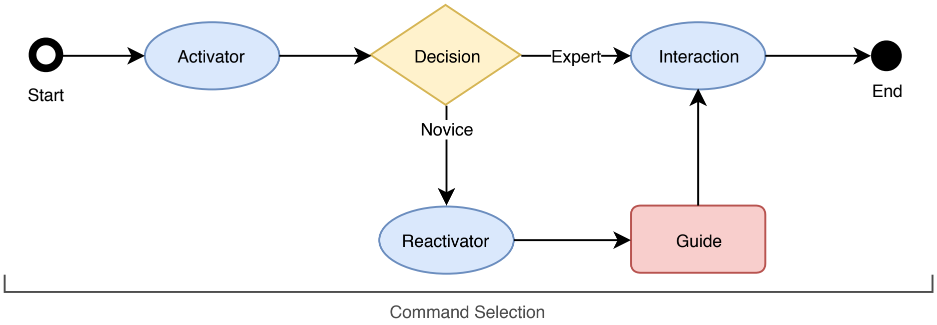


Figure 3.1: A descriptive model of rehearsal-based interfaces. Blue ellipses represent user actions, the yellow diamond is a decision, and the red rectangle is guidance visualization interface. In the model, a user first begins interacting with a rehearsal-based interface by activating it using some input mechanism. Following activation, the user executes a decision to use novice mode or expert mode. If they choose expert mode, they perform the interaction resulting in a command selection. Alternatively, if they choose novice mode, they must first perform the reactivator (the rehearsal interface activator). Once the guidance interface is displayed, the user then performs the interaction which results in a command selection.

After viewing guidance, the novice user performs the same interaction as an expert. The representation of the interaction component as a common part of novice and expert mode usage is a core part of what makes a rehearsal-based interface.

Our model is representative of most, but not all rehearsal-based interfaces. For example, Kurtenbach's original marking menu included an extra "intermediate mode" where a user could delay movement after partially drawing a mark to display guidance [31], and MarkPad had an additional reactivator that only shows a subset of the menu [18]. The model could be extended to incorporate these types of non-essential design enhancements (e.g. intermediate mode could be represented by an arrow connecting interaction back to the decision), but for simplicity we ignore these enhancements and focus on the core components of a rehearsal-based interface.

3.1 Decision to Use Novice Mode or Expert Mode

At some point the user must decide between using novice mode or expert mode. In practice, this cognitive decision is likely made before the activator is triggered. The model captures the point where this decision is executed, meaning the user consciously performs the reactivator to trigger novice mode.

The decision represents a user's intermodal transition. In this section, we contextualise the relevant decision factors from Scarr et al.'s work for rehearsal-based interfaces. The decision to use expert mode reduces to two primary questions: "can I do it?", and "is it worth it?". The first question is a threshold, meaning if the user is unable to perform an expert mode interaction, then no further processing is required and they must use novice mode. This threshold involves two aspects:

- *Knowledge of Command to Interaction Mapping* — In order to perform a command selection, the user must map the command label to the required motor action. This means the user must have learned the correct mapping because most expert interfaces are based on memory retrieval rather than visual search.
- *Motor Action Capability* — A user must be able to perform the motor action in the absence of guidance. For example, when using a marking menu, they must be able to draw a mark in a way that the recogniser understands. They may know what direction to draw to make the right selection, but if they draw the mark at a slightly wrong angle, not long enough, or not straight enough, then they do not have the proper motor action capability.

Once a user is capable of performing an expert selection, there is still no guarantee they will actually use expert mode [35]. A user will compare the pros and cons of using expert and novice mode to create a series of contrasts between the two modes of interaction:

- *Perceived Time Difference* — Although the actual selection time difference between novice and expert modes might be large, a user's perception of the difference is what spurs their decision. Included in the time difference is the cost of identifying and recovering from an error, in the case where they perform an incorrect memory retrieval or a motor error.
- *Confidence* — Expert selections often provide little or no explicit feedback, aside from the result of the selected command. This means the user must be confident in their command to interaction mapping and motor action capability, since it may be hard to detect if the wrong command was issued, so it may take additional time to correct an error.

- *Preference* — Previous studies have shown that people sometimes prefer using expert modes because they feel a sense of achievement [47]. This means their decision may be partially based on challenging themselves to use expert mode simply for the sake of an accomplishment.

A user evaluates the above factors when choosing between using novice or expert mode to perform a selection. If the difference is large enough, then they will perform the selection using expert mode. If not, they will satisfice and continue to use novice mode. Given that the reactivator acts as the switch between using novice or expert mode, it plays a critical role in the intermodal transition for rehearsal-based interfaces.

The two-step decision we describe above is related to the framework for intermodal expertise development presented by Scarr et al. [47]. We provide a more specific instance of their framework, focused on rehearsal-based interfaces. The initial threshold decision relates to Scarr et al.'s exposure and experience component of the initial switch. The second decision involves contrasting different factors, and relates to Scarr et al.'s notion of the user's perception of the new modality, satisficing, and maintenance of the new modality. Our model specifically describes the factors users consider when deciding between the two modalities.

3.2 Reactivator Design

A well-designed reactivator should support both intramodal and intermodal improvement. Reactivators can support intramodal learning by introducing methods that aid the extended learning of the interface. In other words, a reactivator can be designed to not only prepare a user for the transition to expert mode selections, but also encourage that transition.

For example, a reactivator could be designed to require more effort and take advantage of related "effortful" learning benefits [13]. However, as suggested by Kurtenbach [31], the reactivator should not significantly deviate from rehearsal. A deviation can harm the motor learning required to perform the command selection without visual guidance as required for expert mode invocations. A harmful reactivator might alter the learned motor action to create a slower action that propagates from novice performances to expert performances. Consider a reactivator designed to use some special movement gesture, like "shaking" the mouse. This might lead users to learn the expert motor action with the addition of an unnecessary initial gesture. It is important that a reactivator does not harm the ultimate performance of the expert mode as that would reduce the perceived selection time difference between novice and expert mode.

A reactivator can also be designed to manipulate the performance ceiling of novice mode, such as adding a time penalty or delay. Limiting the maximum performance of novice mode prevents satisficing by increasing the relative difference in performance gain for expert mode. Users can also be nudged away from the novice mode by using a slow reactivator that induces some mild frustration, subtly encouraging them to avoid it. However this must be used with caution since it could lead to users avoiding the interface entirely.

Although our focus is on a delay-based reactivator, other forms exist. For example, in Doubletap [49], a single tap reactivates the menu and a double-tap performs a command selection. MarkPad [18] used a key as the reactivator for their intermediate mode. Although these reactivators work, they might not support the intramodal transition to expertise in the same way a delay does.

3.2.1 Delay as a Reactivator

Kurtenbach suggests three reasons a “press-and-wait” delay-based rehearsal activator is effective: the deviation from rehearsal is minimal since the delay does not change the motor action; it is easily avoided when performing expert selections; and novices are not put off by short delays, in fact, it might actually motivate users to begin using expert mode [31]. The delay reduces novice user satisficing by reducing the initial and ultimate performance of novice mode, and this nudges novices towards expert mode. A delay might also allow users to think about the menu locations while waiting, creating a simple form of effortful learning. Given the above, delays longer than those seen in the literature might increase motivation to use expert mode and provide additional time for effortful learning.

Chapter 4

Experiment 1: Effect of Delay

The goal of the first experiment is to determine the effect of delay time on the rate of expert mode use, as well as related measures of learning rate and recall. In the previous chapter we show that a range of delays have been used in rehearsal-based interfaces, yet our model of rehearsal-based interfaces suggests that delay could have a large effect on whether a transition from novice to expert occurs. As a first exploration of this, we manipulate delay in a classic 8-item marking menu [31], arguably the canonical rehearsal-based interface.

4.1 Experiment Description

More specifically, this experiment is designed to answer the research question: Does a longer delay improve the transition from novice to expert and thereby improve the performance of rehearsal-based interfaces? We hypothesise that a longer delay will lead to more expert selections for two reasons: a longer delay adds time for users to consider and think about the locations of menu items, and the delay creates a penalty to encourage expert use. An increase in expert use over time should also counteract the initial frustration caused by waiting for longer delays when first learning the menu item locations. Related to expert use, longer delays should also lead to improved recall of menu item locations. Overall, we expect longer delays to increase the performance of a rehearsal-based interface over a short period of time.

In order to answer our research question, we recruited Amazon Mechanical Turk crowd-workers to perform menu selections using the marking menu with various delay times. We chose delays from the literature (200ms, 333ms and 500ms) and introduced longer delays

(1000ms and 2000ms) to push the limits of how long a delay can be. Each crowdworker used a marking menu with a single assigned delay while completing a sequence of training task blocks designed to measure expert use over time, and multiple blocks of recall tasks to measure the retention of menu item locations. The recall tasks are performed at various periods, including one where crowdworkers returned after 24 hours to measure long-term retention of menu item locations and the impact of relearning. To avoid strong carryover effects that would occur if each participant repeated the same task in multiple delay time conditions, we use a between-subjects design where each participant only completes the tasks with a single delay time.

Why Crowdworkers?

Recruiting crowdworkers for our experiment provides multiple benefits. It made it practical to conduct a between-subjects design with a large number of participants: more than 300 people participated, approximately 60 in each delay condition. Compared to a lab experiment, we also believe using crowdworkers broadens the sample population with a larger age range, more gender balance, and a diverse set of input devices. If an effect is detected under these conditions, it is likely to be even more ecologically valid. Previous studies have shown that crowdsourced experiments and experiments "in the wild" yield valid results [27, 10] and we follow the design recommendations for online experiments suggested by Kittur et al. [26] to ensure high quality data.

One concern is that crowdworkers may "cheat" by recording the menu item locations in some offline way, such as writing them down or using a screen capture. We employed several countermeasures to discourage and detect this.

After each crowdworker's task was approved and they received payment, we asked if they captured or recorded the menu item interface in order to remember the mapping. By asking the question after payment, and not suggesting doing anything to remember the mapping was necessarily wrong, we could elicit an honest answer and remove those participants from the dataset. In fact, only 3 answered yes. We also analysed our data to look for cheaters, since performance profiles would appear as an obvious outlier. None were found. In addition, we monitored key crowdworker forums such as *Turkopticon* to discover if any cheating method was discussed or suggested for our task. We also performed internet searches of our requester ID to find other posts or blogs discussing our task. We did not find any evidence of cheating in those forums or searches. Perhaps most importantly, our experiment dissuaded cheating by paying a good wage, maintaining a reasonable task duration, and according to feedback from multiple participants, it was even enjoyable compared to usual crowdsourcing tasks.

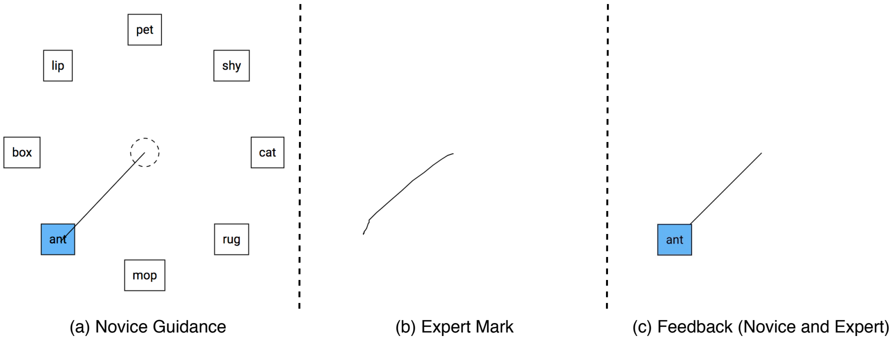


Figure 4.1: Marking menu interaction and interface: (a) novice mode shows a guidance interface with each menu item radially arrayed around the initial input position, the mark to select an item ("ant" in this example) is shown as a beautified straight line, the dashed circle indicates the size of the deadzone but was not visible to participants; (b) an expert mode mark is performed without the guidance interface, the actual path of the input device is rendered; (c) after selecting an item in either mode, only the selected item is shown for 500ms.

4.1.1 Menu Interface Design

Throughout the experiment, participants use a standard single level marking menu with 8 items radially distributed at standard compass directions (Figure 4.1). The menu was designed according to suggestions from Tapia and Kurtenbach [51]. An expert mode selection is made by beginning an input action (e.g. pressing the mouse button) and then drawing a mark in one of the 8 compass directions without guidance (Figure 4.1b). Novice mode selections are made by beginning an input action, waiting for the delay time to complete (the reactivator), viewing the guidance interface showing menu item locations, then drawing a mark using the same interaction as the expert mode (Figure 4.1a).

To implement a delay reactivator, there must be a method to determine when the input point (e.g. the mouse cursor) is not moving and the delay timer is running. We use a distance threshold from the menu activation point, realised as a circular "deadzone" with a 17 pixel radius (10% of the guidance interface radius). This is similar to the "activation distance" used by Pook et al. [45]. If the input point remains inside the deadzone until the delay timer completes, then novice mode is activated. If the input point exits the deadzone before the delay time completes, the delay timer is cancelled and the user is using expert mode.

We use this deadzone approach for its simplicity and predictability. Using a threshold

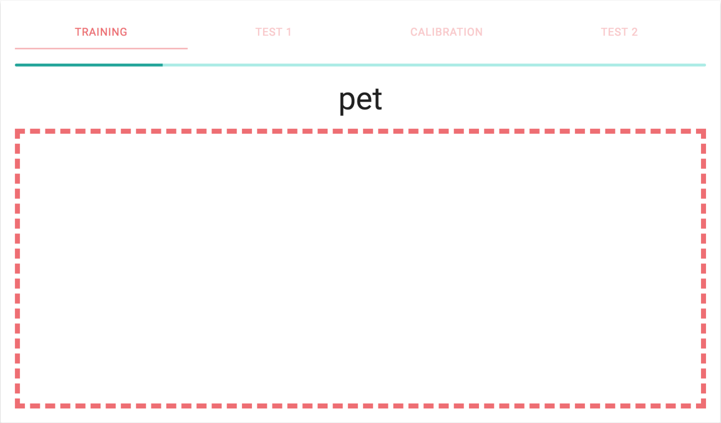


Figure 4.2: Experiment interface. Across the top participants see which section of the experiment they are currently in (the “training” task in this example). Beneath that, a progress bar indicates how much of the current section has been completed. The dashed red area is the space in which participants can perform a menu selection. Above this area is the current stimulus (the word “pet” in this example) to indicate which menu item to select.

on total relative input movement distance would accumulate small input jitter, characteristic of touch input devices like touch pads. Our deadzone approach also rules out the “intermediate mode” used in some marking menu implementations. However, this method reveals novice guidance by pausing after completing a partial mark, which would be a confound in our experiment context.

Menu feedback is provided during a selection and following a selection, in the same way demonstrated by Kurtenbach [33]. During an expert selection, the mark is displayed exactly as the user draws it. However, during a novice selection the mark is corrected to be a perfectly straight stroke. After completing a selection in either novice or expert mode, a beautified straight mark to the selected menu item, and the selected menu item from the novice guide, are displayed for 500ms (Figure 4.1c). An accompanying video demonstrates all menu interfaces and experiment tasks.

4.1.2 Tasks

The marking menu was used in a *training* task, *recall* task, and *distractor* task. Each task is performed in a common experiment interface (Figure 4.2). The interface communicates progress through the experiment, presents the stimulus menu item for the current task trial, and provides a defined area to activate the menu as a dashed red rectangular area.

Training Task

In the *training task*, participants must make selections using the marking menu with their choice of using novice or expert mode. Participants receive a stimulus and must perform a correct selection using the marking menu before proceeding to the next trial. The screen briefly flashed red if the selection was incorrect, and the participant had to complete the correct selection to continue. This ensures crowdworkers do not rush through the task trials, and resembles a real application where users must try again to choose a desired item.

Recall Task

The *recall task* is similar to the training task, with participants prompted with a menu item stimulus and then selecting that item using the marking menu. However, novice mode guidance and post selection feedback were disabled and no indication is given whether the selected item matched the stimulus. Feedback after a selection is an empty box indicating only that the system registered their selection without telling them which menu item was selected. Unlike the training task, the experiment proceeds to the next task regardless if the selection was correct. The task is designed to minimise learning and test the participant's retention of the menu item locations.

Distractor Task

In the *distractor task*, the stimulus and marking menu items are arrows pointing in the corresponding compass direction. This means participants no longer have to think about the menu locations and instead make a selection bounded only by stimulus reaction time and the motor action. The task is intended to distract users from the menu locations used in the training, and provide a more realistic context for recall tasks. Additionally, it provided us a measure of how fast users could make expert selections since they did not have to recall the menu item location. We used this data to verify that users can easily perform expert selections in under 200 ms, supporting this as a reasonable lower bound for controllable delay time with a marking menu.

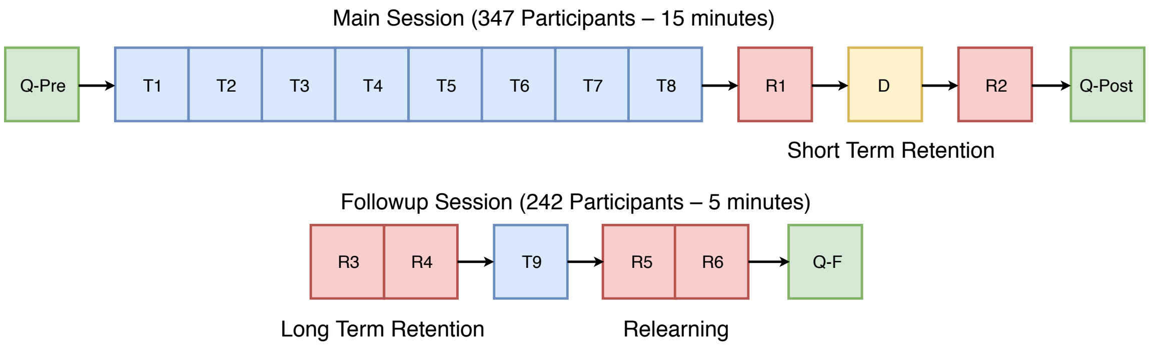


Figure 4.3: Experiment 1 procedure. Green blocks prefixed with Q indicate a questionnaire, blue blocks prefixed with T are training blocks, red blocks prefixed with R are recall blocks, and the yellow block beginning with a D is a distractor block. Between consecutive blocks a 3 second rest is enforced. Before beginning each new section of the experiment, the participant completed a short tutorial to familiarise them with the interface.

4.1.3 Protocol

The study consists of two sessions, a 15-minute *main session* for training and short term recall tests, and a 5-minute *follow-up session* scheduled for 24 hours later to test both long term retention and relearning. Measuring retention after 24 hours follows Anderson and Bischof's suggestion [1] for testing cognitive learning. Note they also suggest a transfer task to test motor learning, but the motor action in our task is simple and not our focus. Figure 4.3 provides an overview of the protocols for the main session and follow-up session.

The *main session* begins with a pre-questionnaire collecting participant demographics, handedness, and pointing device information (mouse, trackpad, other).

The participant is next given interactive instructions on how to use the marking menu. The instructions mimic the task interface, providing a dashed red area for menu selections. Before proceeding to the measured tasks, the participant must make 3 selections with novice mode and another 3 selections with expert mode. We only instruct participants on how to use the menu and do not give any task specific instructions such as trying to remember the menu items, complete the task quickly, or minimise errors. We did this to avoid biasing the participant's motivation so the task captures more natural behaviour. It is up to each participant to decide between using the novice and expert modes of the menu based on the decision criteria we outlined in our model earlier.

Following the instructions, the participant completes 8 blocks of 24 training tasks.

Between each block a 3-second resting period is enforced, the rest slows the pace of the experiment. After these training blocks, the participant completes recall task trials for each of the 8 items. The participant then completes 4 blocks of 8 distractor tasks, which takes approximately 1 minute. Retention is tested once more following the distractor task with another 8 recall tasks trials, one for each menu item. Before they complete the recall and distractor tasks, the participant first completes a tutorial ensuring they understand the menu will not be displayed. In this session, only one recall task block is tested at a time in order to limit the amount of learning the participant gains from being tested. The session ends with a questionnaire based on the NASA Task Load Index (NASA-TLX) to measure perceived workload [25] in six dimensions. Of special importance to our experiment, is the dimension of perceived frustration, an aspect of workload that may increase with longer delays. At the end of this main session, the participant is paid and invited to the follow-up session.

The *follow-up session* occurred approximately 24 hours after the main session. It begins with 2 consecutive blocks of recall tasks to test long term retention. We added a second block of recall tasks to test the participant's confidence of selections. A participant who is confident in their selection will answer with the same selection each time. Participants then complete a single training task block of 24 menu selections, exactly the same as a training block in the main session. The intention for this single training block was to determine if the participant could relearn the menu items. A final two recall task blocks measure whether this short opportunity for relearning had an effect on recall. After all tests are completed and the participant was paid, a final questionnaire asked if they used any memory aids to recall menu item locations, such as taking a screen capture of the open menu or writing down notes. Our goal was to detect "cheating" in the context of our experiment, but we did not phrase the question in a negative way or use the term cheating. As far as the participant knew, the goal of our study was to understand if people used memory aids.

4.1.4 Design

The experiment used a between-subject design with DELAY as the independent variable. We chose 5 delays: 200MS, 333MS, 500MS, 1000MS and 2000MS. 200MS was chosen to be the shortest delay that users could reliably escape the menu's dead zone to perform an expert selection. We initially established this minimum delay time through trial and error and small pilots. Pin-and-Cross [41] and FastTap [24] also use a 200MS delay. 333MS was chosen to be representative of current delay lengths and the original marking menu delay [28]. 500MS was used in MarkPad [18]. 1000MS and 2000MS were chosen to test the limits of delay length. Our delay selections are representative of commonly chosen delays in addition to new, even longer delays.

Dependent measures from the training task were:

- *Successful Expert Rate*, the proportion of trials completed using a single correct expert mode selection.
- *Number of Errors*, the total number of incorrect selections;
- *Selection Time*, the duration of time from the beginning of a selection (start of menu activation) until the selection is completed (after the menu interaction).

The recall task had two dependent measures:

- *Recall Rate*, the proportion of correct selections.
- *Confidence*, the proportion of recall task selections where the participant responded the same in each paired block.

Recall can be broken down into four types: instantaneous recall measured immediately after all main session training task blocks (block R1 in Figure 4.3), short term recall measured after the 1-minute distractor task (block R2), long term retention measured as the first step of the follow-up session on the following day (block R3, with block R4 to test confidence), and recall after relearning, measured after an additional training block in the follow-up session (block R5, with block R6 to test confidence).

Menu Items, Frequency, and Ordering

The menu items are: mop, cat, lip, rug, box, shy, pet, ant. These were chosen from Scarr et al.'s study on spatially stable interfaces [46] such that all items have three letters and none begin with the same letter. To avoid stimulus order effects, each participant received a menu with a randomised mapping of item label to item location.

Previous studies have shown that realistic command use follows a Zipfian distribution [21, 11, 15, 14]. Prior work uses a Zipfian distribution to mimic realistic command use in experimental settings. Liu et al. studied the effect of different frequency distributions using two Zipfian distributions and a uniform distribution [40]. Their results suggest that selection time depends on the frequency distribution. As a result, in training task blocks we present stimuli in a Zipfian distribution over eight menu items to better represent realistic command usage frequencies. Each menu item has a rank based on its position in the Zipfian distribution of frequencies we used: 9, 4, 3, 2, 2, 2, 1, 1. This means the menu item with rank 1 appeared 9 times in a block, the item with rank 2 appeared 4 times, the item with rank 3 appeared 3 times, and so on. In total, each block has 24 menu item selections. We randomly assigned item labels to item ranks across participants.

Stimuli order was randomised in such a way that no two menu items were displayed twice in a row. The order stimuli appear in the recall tasks was also randomised for each test and each participant. Note the recall task blocks did not use a Zipfian distribution, the 8 stimuli were each presented once in a uniform random order.

4.1.5 Participants and Apparatus

We recruited 347 crowdworkers from the United States using Amazon Mechanical Turk. Participants received $2 for the main session and an additional $1 if they returned to the follow-up session. Since the main session took less than 15 minutes on average, and the follow-up session less than 5 minutes on average, these payment rates are approximately minimum wage and considered fair payment to crowdworkers. The experiment was developed using JavaScript and managed by CrowdCurio [55]. Only participants using Chrome or Firefox browsers could participate to reduce the possibility of browser compatibility bugs.

4.2 Results

Of the 347 participants, 152 were female, 193 male, and 2 preferred not to answer. The mean age across all participants was 33.8 (SD = 9.3). 242 (70%) of the original 347 participants returned for the follow-up session. In terms of how they interacted with the computer, 17 participants used their left hand to control their pointing device, 274 participants used a mouse, 71 used a trackpad (i.e. touchpad), 1 participant used a touch screen, and 1 participant used a trackball. We tested for an effect of device type on all dependent measures and found no effect.

In terms of cheating, we removed 3 participants who answered ”yes” to the post-payment question asking if they used an offline method to remember menu items. We also filtered for a pattern indicating cheating, namely expert use greater than 95% with an error rate below 5%, but did not find any matches.

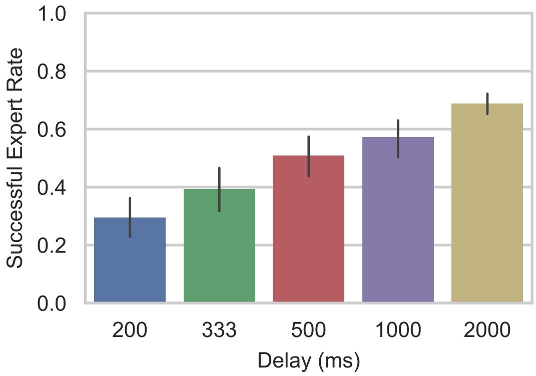


Figure 4.4: *Successful expert rate* by DELAY. Results aggregated across all training blocks in the main session (in all graphs, plotted values are means with error bars 95% CI, unless noted otherwise).

We also removed a total of 10 participants (3%) determined to be outliers or had technical errors. One participant's data was removed due to an obscure bug that allowed them to do the experiment twice, 2 were removed for not completing the first session in a reasonable time (both more than 20 minutes), and 7 were removed for making more than 77 errors, which was more than 3 standard deviations from the mean. After removing these participants, we had condition sample sizes of 62 in 200MS, 63 in 333MS, 69 in 500MS, 65 in 1000MS, and 73 in 2000MS.

In the analysis to follow, all data was tested for normality and sphericity. An ANOVA with Tukey HSD post hoc tests was used, unless noted otherwise. A one-way ANOVA was used when only analysing a factor of DELAY. When analysing DELAY × BLOCK we used a two-way mixed factorial ANOVA with BLOCK as a within-subject factor and DELAY as a between-subjects factor.

4.2.1 Successful Expert Rate

Successful expert rate is the proportion of training task trials completed using only a single correct expert mode selection. The rate is computed per training block, creating one successful expert rate data point per block, per participant. Results include only the 8 training blocks in the main session (T1 - T8), unless noted otherwise.

In general, longer delays lead to higher rates of successful expert mode selections (Figure 4.4). We found a main effect of DELAY on successful expert rate ($F_{4,328} = 24$, $p < 0.001$,

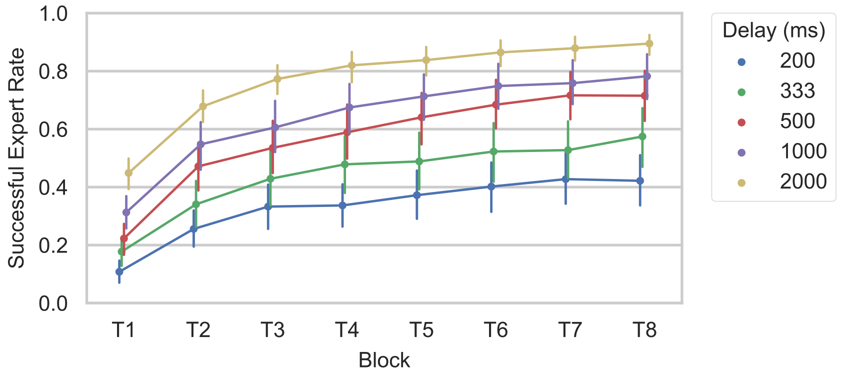


Figure 4.5: *Successful expert rate* for each DELAY by training blocks T1 to T8 in the main session (see Figure 4.3 for session and block progression).

$\eta_p^2 = 0.23$). Post hoc tests found no significant difference for adjacent delays (i.e. 200MS and 333MS), however all other differences between non-adjacent delays were significant ($p < 0.01$). Participants in the 2000MS condition made 18% more expert selections than those in the 500MS condition, and 40% more than those in the 200MS condition. Note that the standardised effect size[1] of .23 is also well above medium (.13), and approaching what is considered a large effect (.26) [5].

A longer delay appears to increase expert selection rate by a constant factor, and the learning curve of expert selection rate over main session blocks follows an exponential curve (Figure 4.5). In order to further analyse the characteristics of the learning curve, we regressed aggregated data points per delay for each *block* using the following exponential function f:

$$expertuse = f(block) = a \cdot e^{-b \cdot block} + c \tag{4.1}$$

Note that in order to make the y-intercept represent performance after one block, the block parameter is zero-based, meaning our experiment training block T1 is represented numerically as $block = 0$, and so on.

The fitted model parameters (a, b, and c) are provided in Table 4.1 with R^2 values for all fitted delay curves. Notably, R^2 are all above .98, demonstrating that learning is exponential. The curves are similar to exponential learning curves based on success rate [38]. The parameters a and b indicate all curves have a similar shape, but the parameter

[1] All standardised effect sizes are generalised eta squared [44, 5]

Table 4.1: Exponential regressions of *expert use* by *block* for each DELAY using Equation 4.1. The y-intercept is calculated to aid interpretation, and $f(7)$ and $f(100)$ are calculated expert use rates for block 7 (i.e. block T8) and a hypothetical block 100 to show near-asymptotic characteristics.

DELAY	R^2	a	b	c	y-intercept	$f(7)$	$f(100)$
200	0.98	-0.31	0.53	0.43	0.11	0.42	0.43
333	0.99	-0.38	0.51	0.56	0.18	0.55	0.56
500	0.98	-0.49	0.50	0.72	0.24	0.71	0.72
1000	0.99	-0.45	0.53	0.78	0.32	0.77	0.78
2000	0.99	-0.43	0.68	0.88	0.45	0.88	0.88

c monotonically increases by 6 to 16 units as delay increases. By calculating the y-intercept created by each curve, we can see how the rate of expert use monotonically increases as delay increases. The shape of the curves also shows how the performance is nearly asymptotic at the last block, and is unlikely to improve further.

As a thought experiment, we calculated the rate of expert use for the last training block on the first day using $f(7)$ (recall, block T8 is represented as $block = 7$ in Equation 4.1) and a hypothetical one hundredth block $f(100)$ for all delays. The last two columns of Table 4.1 provide these estimates. The pattern suggests the rate of expert use for shorter delays will reach an asymptote, resulting in a lower ultimate expert usage rate. In other words, the overall increase in expert usage with longer delays appears to persist compared to shorter delays, even after a long period of use.

4.2.2 Recall Rate

Recall rate is the proportion of correct selections during the recall task in which no guidance or item feedback was provided. The rate is computed per recall block which always had exactly 8 trials, there is one data point per block, per participant. Recall rate was measured at four points during the experiment (see also Section 4.1.4 and Figure 4.3). Short term retention was measured immediately after all training blocks were completed (block R1) and again after a 1-minute distractor task (block R2). Long term retention was measured after 24-hours in two parts: before and after a single training block (T9) to also examine the effect of relearning on recall. In addition, each part of long term retention measurement had two consecutive recall blocks. We use the second blocks (R4 and R6) to measure confidence.

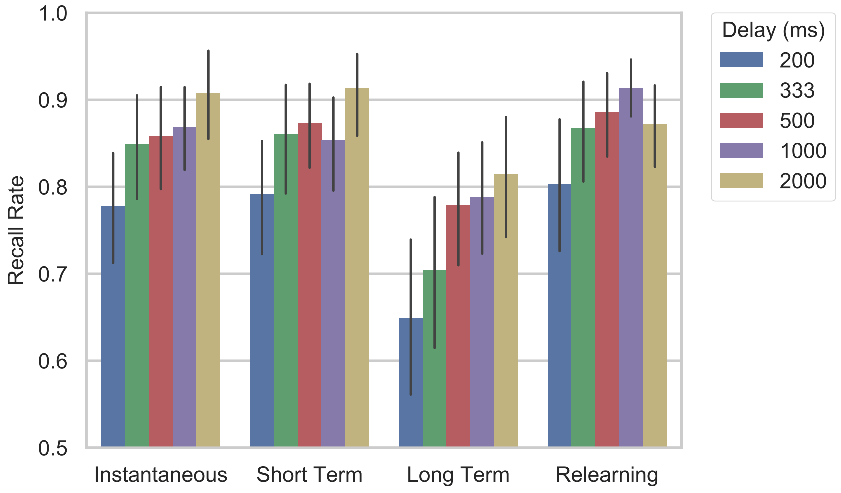


Figure 4.6: Recall rates at various measurement times. During the main session, *instantaneous* recall (R1) occurred directly following the training blocks and *short term* recall (R2) occurring after a distractor task. During the follow-up session 24 hours later, *long term* retention (R3) was measured upon entering the session and *relearning* (R5) after performing a single training block.

Overall, there was a general trend where participants remembered 1 less item in the 200ms condition compared to longer delays (Figure 4.6).

Instantaneous Recall — A one-way ANOVA examining delay on instantaneous recall (measured using block R1) indicates a main effect of delay ($F_{4,328} = 2.8$, $p < 0.05$, $\eta_p^2 = 0.03$), and post hoc tests found a significant difference between 200ms and 2000ms ($p < 0.05$). Participants in the 200ms condition remembered approximately 1 less menu item.

Short Term Recall — A one-way ANOVA of delay on short term recall (block R2) also indicates a main effect of delay ($F_{4,328} = 2.6$, $p < 0.05$, $\eta_p^2 = 0.031$). Post hoc tests indicate a significant difference between 200ms and 2000ms ($p < 0.05$). Again, participants in the 200ms condition remembered approximately 1 less menu item.

Long Term Retention — A one-way ANOVA examining delay on long term retention (block R3), also has a main effect ($F_{4,237} = 2.9$, $p < 0.05$, $\eta_p^2 = 0.05$). Post hoc tests indicate a significant difference between 200ms and 2000ms ($p < 0.05$) where once again participants in the 200ms condition remembered approximately 1 less menu item. Regarding their selection confidence, a one-way ANOVA of delay on the proportion of responses

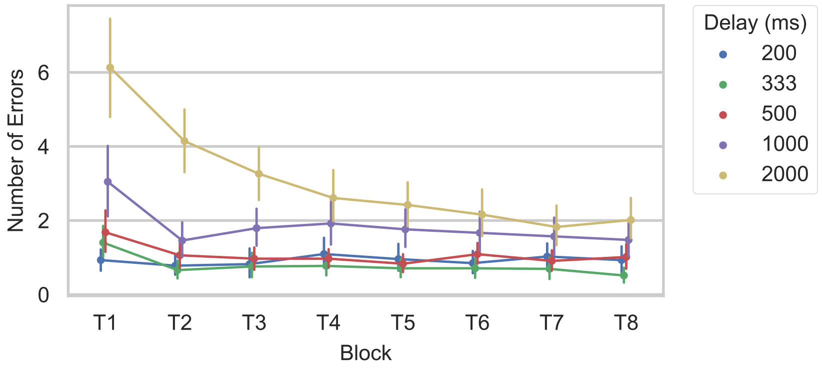


Figure 4.7: Experiment 3: Number of trials containing an error by BLOCK and DELAY.

that remain the same across blocks R3 and R4 found no significant effects, suggesting participants were not guessing.

Recall after Relearning — After participants completed relearning training (block T9), there was no main effect of DELAY ($F_{4,237} = 1.9$, $p = 0.1$, $\eta_p^2 = 0.03$), however we can see a similar trend to the other three measurements of recall. The data for recall following relearning has a similar trend to the previous recall measurements. A one-way ANOVA of DELAY on recall confidence for blocks R5 and R6 found no significant effects, suggesting again that participants were not guessing.

4.2.3 Number of Errors

For lower delay times, we observed a general trend where errors are constant and low, however longer delays result in more errors initially, but quickly reduce and level off (Figure 4.7). We calculated the total number of trials per block in which at least one error occurred. A two-way ANOVA of DELAY × BLOCK on number of errors found a significant effect of DELAY ($F_{4,328} = 22.6$, $p < 0.001$, $\eta_p^2 = 0.11$), BLOCK ($F_{1.6,532.5} = 49.6$, $p < 0.001$, $\eta_p^2 = 0.08$), and a significant interaction after correcting for sphericity ($F_{6.5,532.5} = 14.4$, $p < 0.001$, $\eta_p^2 = 0.09$). The interaction is likely due to the close grouping of delays other than 2000MS. Post hoc analysis found that 2000MS had more errors than all other delays, and 1000MS had more errors than all other delays, except between 2000MS and 500MS (all $p < 0.05$).

4.2.4 Selection Time

The time taken to complete a selection is significantly faster for participants in the 2000ms condition (Figure 4.8). Selection time is measured as the time from activating the menu until the selection is completed. Only main session training blocks are included (blocks T1 to T8). We calculate the median selection time for each participant by block in order to reduce the skew created by outliers. A one-way ANOVA of DELAY on selection time was significant ($F_{4,272} = 5.2$, $p < 0.001$, $\eta_p^2 = 0.07$). Post hoc analysis indicates a significant difference between 2000ms and all other delays except 1000ms (all $p < 0.05$).

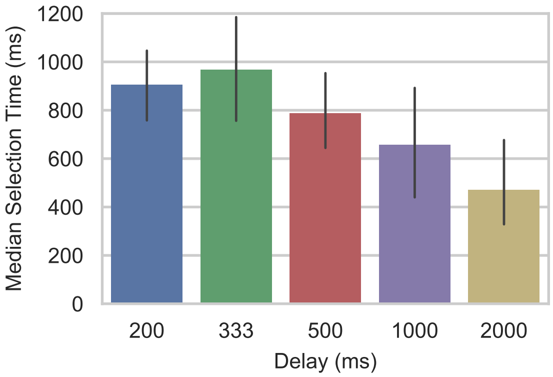


Figure 4.8: Experiment 3: Median selection time by DELAY for all main session training blocks.

4.2.5 Perceived Workload

How people perceive waiting for very long delays is an important aspect. Most importantly, we expected frustration to be a factor, but we found no statistical support for increased frustration with longer delays. A questionnaire based directly on NASA-TLX captured ratings in six dimensions of perceived workload: *Mental Demand*, *Physical Demand*, *Temporal Demand*, *Performance*, *Effort*, and *Frustration*. To simplify the questions and user interface for crowdworkers, we reduced the 20-point NASA-TLX scale to a numeric 7-point scale, where 1 is “very low”, and 7 is “very high”. Results for each dwell condition were compared using Kruskal-Wallis tests, since not all measures were normally distributed. Values are reported as mean and standard deviation since the scale is interval.

We found no significant differences for any subjective measure (Table 4.2), tests range from ($\chi^2(4) = 6.82$, $p = 0.145$) to ($\chi^2(4) = 0.953$, $p = 0.92$). Foremost, *Frustration* was

Table 4.2: (Experiment 1) Mean and standard deviation of NASA-TLX results.

	Mental	Physical	Temporal	Performance	Effort	Frustration
200	4.5 (1.6)	2.8 (1.5)	3.5 (1.7)	5.3 (1.4)	5.5 (1.4)	2.9 (1.8)
333	4.7 (1.3)	2.9 (1.5)	3.4 (1.5)	5.6 (1.3)	5.4 (1.5)	2.5 (1.6)
500	4.6 (1.5)	2.7 (1.7)	3.8 (1.5)	5.6 (1.3)	5.7 (1.3)	2.8 (1.8)
1000	4.7 (1.5)	3.1 (1.7)	3.4 (1.4)	5.5 (1.1)	5.8 (1.1)	3.1 (1.7)
2000	4.5 (1.4)	2.5 (1.6)	3.3 (1.7)	5.8 (1.2)	5.7 (1.3)	2.7 (1.5)

similar for all dwell times: the overall mean rating was below neutral at 2.8 (SD = 1.7), ratings by condition ranged from 2.5 (SD = 1.6) for 333ms to 3.1 (SD = 1.7) for 1000ms. Most participants felt that they performed well, rating *Performance* as 5.6 (SD = 1.3) overall. In addition, participants reported a high level of overall *Effort* (5.6, SD = 1.3) and *Mental Demand* (4.6, SD = 1.5) , but below medium *Physical Demand* (2.8, SD = 1.6). The experiment was also considered just above neutral for *Temporal Demand* (3.5, SD = 1.6).

4.3 Discussion

Our main finding is that a longer delay leads to more expert selections with a rehearsal-based interface, and this is achieved with lower average selection times over the training period, and no detectable difference in perceived frustration.

For example, a long delay of 2000ms yields 39% more successful expert selections than shorter delays like 200ms or 333ms used for several rehearsal-based interfaces proposed in previous work. The increased use of expert mode with longer delays of 1000ms or more also resulted in significantly faster overall selection times. This means that the initial large time penalty for triggering novice mode does not overshadow the time savings when more expert mode selections are used later. Given the increased number of expert selections with longer delays, we would expect the recall rate to improve with longer delays too. However, recall rate was only significantly different between an imperceptible 200ms delay and the longest 2000ms delay.

Considering Item Frequency

The frequency of menu items in test blocks follows a non-uniform, near-Zipfian distribution to approximate the distribution of real command usage [21]. This means our analysis for expert use, time, and error implicitly weighs frequently appearing menu items more than less frequent ones. Even recall may be affected since remembering frequent items may be more likely. We believe combining all frequencies in the analysis above is the best representation of how any effects are perceived in a real system. However, for completeness, we also tested for interactions between frequency (at levels 9, 4, 3, 2, and 1) and delay for all dependent variables. We found a similar pattern of significant differences, and the same relative ordering of delays, for all measures, with minor differences. Expert use between 500ms and 2000ms was not significant for frequency 1, and there were additional significant differences for frequency 2 between 1000ms and 2000ms and frequency 3 between 333ms and 500ms. Also, recall for 2000ms and 200ms in significantly different for frequencies 1, 2 and 3, but not for frequencies 9 and 4.

Long Delays May Not be Frustrating

We found it remarkable that long delays did not significantly affect the perceived workload, especially the level of frustration. We assumed waiting for a long delay like one or two seconds would have some measurable effect. Perhaps this is because users of longer delay times make less use of the novice mode overall, so the long delay times are experienced only at the beginning. In other words, the benefit of completing the task with more expert selections in the final training blocks may overshadow any frustration caused by waiting for long delay times in initial training blocks. One might argue that rating frustration near the beginning of training would have found an effect, but since the experiment was short (about 15 minutes), any pronounced feeling of frustration would surely have persisted until the questionnaire. Regardless, assessing frustration after achieving some benefit is more ecologically valid and a more relevant measure of the total experience. Moreover, past study designs like Cockburn et al. [12] also assess frustration after the experiment.

To test the theory that overall wait times were comparable across delay times when rate of expert use is considered, we compute the average time spent waiting for a novice mode delay as $(1 - SER) * delay$. Where SER is the mean *successful expert rate* overall, for the entire training period.

We find that users of a 2000ms delay menu spend 622ms waiting on average, while users of the 200ms menu wait 116ms. Given how poorly people perceive time, a 500ms difference may not have been noticeable. Note this calculation assumes the perceived cost of making an error is equivalent to the alternative cost of waiting for the novice mode delay. We also discuss this below.

Another theory for the lack of a frustration effect is that, since long delays forced a switch to expert mode, our participants ultimately felt a greater sense of achievement in memorising the items [47]. This ultimate sense of achievement may also have eclipsed any initial frustration due to long delay times. However, our study did not assess sense of achievement directly.

An Imperceptible Delay Reduces Recall

A very short 200ms delay can harm long term retention, but overall, longer delays do not significantly increase long term retention. One reason may be that waiting for 200ms is not enough time to expend cognitive effort to even try to remember the item location. Another reason may be related to how actions can be combined into "microstrategies", where users perceive multiple actions as being a single action [20]. We suspect that users of shorter delays perceive the invocation and the delay as a single action, whereas longer delays are perceived separately from their invocation.

Cost of Error versus Waiting

Our initial expectation was that an increase in successful expert use could be directly attributed to remembering more menu item mappings. Although our analysis of recall rates showed some effect, there is not enough to plausibly be the sole factor.

Another possible explanation is how people perceive the cost of errors versus waiting for novice mode. In most applications, errors result in additional time to perform corrective actions, and this usually results in more frustration. Our results indicate that users of longer delay times were not more frustrated, and do not spend more time per trial. This could indicate that users of longer delays perceive the cost of waiting for the delay to be costlier than just "going for it" and making a selection even if this sometimes results in an error. More errors in the first training block when using longer delays supports this theory ($F_{4,328} = 19.6$, $p < .001$, $\eta_p^2 = .19$, with significant differences between 1000ms, and 2000ms and all other delay lengths ($p < .05$). However, it is important to note a limitation in our simplified task where the cost of a trial error is minimal. This is similar to recovering from an application error with a quick "undo" shortcut key, but some application errors would be much costlier.

Confidence Could Still Be a Factor

Although we did not find significant differences for our recall confidence measure, confidence may still play a part. Our measure assumed less confident users would change some selections over two consecutive recall test blocks, but perhaps this is more apt to measure the degree of guessing. A user who achieves some learning can be less confident in their knowledge of item locations, but still make consistent item selections during the recall tests. It is possible that our participants in the shorter delay conditions were less confident in their recall, and that confidence factored into whether or not they make more expert selections during training.

Impact on Intermodal Learning

Delay appears to impact two of Cockburn et al.'s [12] intermodal learning factors. *Initial learning* appears to be impacted by delay, as shown in the first block users of longer delays (such as 2000ms made many more successful expert selections). Furthermore, the *ultimate performance* of users of long delays was also impacted by delay as these users performed many more successful selections after extended training. In order to truly measure ultimate performance, a longitudinal study spanning multiple days or more is required.

Of course, these results are specific to a marking menu. In the following experiment, we generalise them to another type of rehearsal-based menu called Fast2Click based on FastTap [24]. We also suspect non-significant recall differences may be due to a ceiling effect when memorising only 8 items. Adding more menu items would test this theory, but adding more than 8 items to a marking menu requires a hierarchical structure [30] which introduces a more complex two-stage visualisation and motor action. Fast2Click can scale to more items without introducing more complexity, so once we verify that our results generalise to Fast2Click, we test a 15-item Fast2Click menu in a third experiment.

Chapter 5

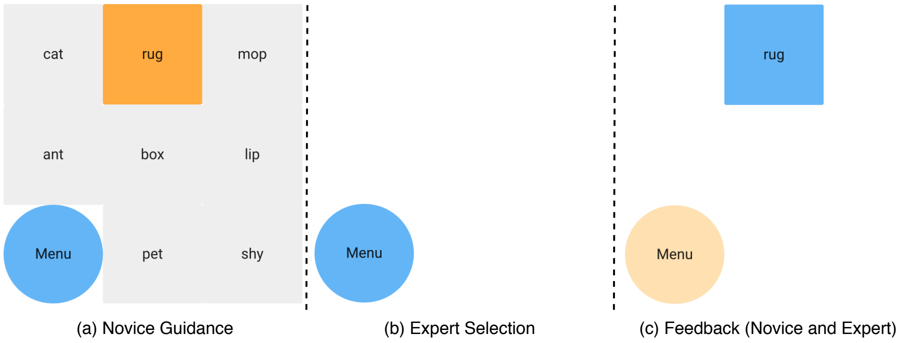
Experiment 2: Generalisation

The goal of this experiment is to generalise our delay time results to another rehearsal-based menu. This study was conducted on Amazon Mechanical Turk using the same apparatus as the previous experiment, and follows the same protocol.

5.1 Experiment Description

5.1.1 Menu

We designed a Fast2Click menu for participants to use instead of a marking menu (Figure 5.1). Fast2Click is based directly on the FastTap [24] multitouch menu, which has a novice and expert selection mode that follows Kurtenbach's principle of rehearsal [31]. Novice selections are made by clicking an always-displayed ”menu” button and pausing for the given delay without moving the mouse outside the button. After the delay completes, a novice visualisation consisting of a 3 × 3 grid of items is shown. Clicking on an item completes the selection. Expert selections are made by clicking the ”menu” button and then, without pausing, clicking on the position of the desired item (as if the item grid was visible). Novice mode cannot be reactivated after exiting the ”menu” button until the selection is cancelled or completed. Selections can be cancelled by clicking outside the menu grid. The ”menu” button acts like the marking menu deadzone, and it is round to make it direction independent during pointing movements.



(a) Novice Guidance (b) Expert Selection (c) Feedback (Novice and Expert)

Figure 5.1: Fast2Click menu interaction and interface: (a) novice mode shows a guidance interface with each menu item arrayed in a grid, the item being hovered is highlighted ("rug" in this example); (b) an expert mode selection is performed without the guidance interface; (c) after selecting an item in either mode, only the selected item is shown for 500ms.

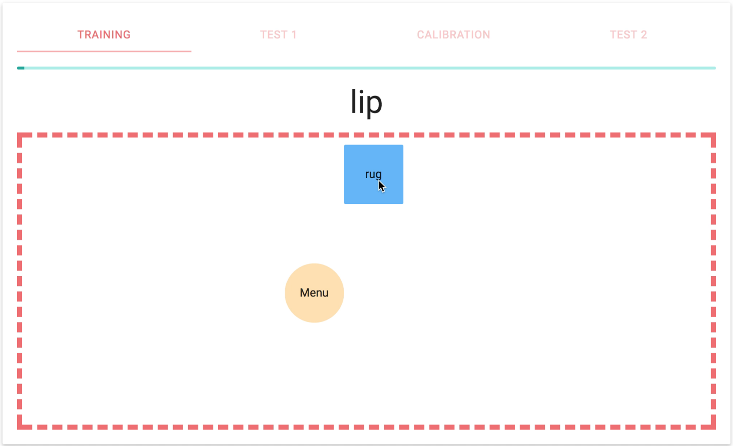


Figure 5.2: Experiment 2 interface. Across the top participants see which section of the experiment they are currently in (the "training" task in this example). Beneath that, a progress bar indicates how much of the current section has been completed. The dashed red area is the space in which participants can perform a menu selection. Above this area is the current stimulus (the word "lip" in this example) to indicate what menu item to select.

5.1.2 Tasks

The experimental tasks were identical to those used in the first experiment except the distractor task (Section 4.1.2, Figure 5.2). Rather than arrows the menu used word labels such as "Top Left" and "Bottom Right". Initial tests suggested directional arrows were not as clear for a grid layout.

5.1.3 Protocol

The experimental protocol was identical to experiment 1 (Section 4.1.3).

5.1.4 Design

The study design was identical to the previous experiment (Section 4.1.4) with the exception that only two different delay levels are tested. We chose 500MS as the shortest delay because in a pilot experiment, participants were unable to consistently make selections with Fast2Click expert mode with a delay of 333ms or 200ms. 2000MS was chosen in order to observe the extreme effects of longer delays. Testing two delay levels is sufficient to show the general effect of delay.

5.1.5 Participants and Apparatus

57 workers were recruited using Amazon Mechanical Turk, all from the United States. They were paid $2 to complete the experiment and an additional $1 if they returned to the follow-up session., the same as in experiment 1. Note that participants who took part in the first experiment were not allowed to do this experiment. The apparatus was the same as the first experiment.

5.2 Results

Of the 57 participants, 35 were female, 22 male, and 1 participant preferred not to answer. 37 of the participants returned for the follow-up session (60%). The age distribution was similar to the previous experiment with a mean age of 36.5 (SD = 10.4). Only 1 participant used their left hand to control their pointing device. 45 participants used a mouse and 13 used trackpads.

Table 5.1: (Experiment 2) Exponential regressions of *expert use* by *block* for each DELAY using Equation 4.1. The y-intercept is calculated to aid interpretation, and $f(7)$ and $f(100)$ are calculated expert use rates for block 7 (i.e. block T8) and a hypothetical block 100 to show near-asymptotic characteristics.

DELAY	R^2	a	b	c	y-intercept	$f(7)$	$f(100)$
500	0.99	-0.43	0.61	0.81	0.37	0.80	0.81
2000	0.99	-0.34	0.99	0.94	0.60	0.94	0.94

We removed two participants who said they captured a screenshot of the menu. Two other outlying participants were also removed: 1 for making 218 errors which was 3 standard deviations from the mean and 1 for a technical error. After removing these participants, the 2000MS condition had 28 participants and 500MS had 25 participants.

5.2.1 Successful Expert Rate

Again, a longer delay led to higher rates of successful expert use (Figure 5.3). When analysing successful expert use, we found some data failed to pass Levene's test of equal variance. Due to the unequal variance between conditions we use Welch's t-test as it does not assume equal group variances. The t-test shows a significant difference in successful expert rate over the entire experiment between 500MS and 2000MS ($t_{27.2} = -3.8$, $p < 0.001$). This is an increase of approximately 17% for the 2000MS condition, which is comparable to the 20% increase between the same conditions using marking menus. Participants with a 2000MS delay had a mean successful expert rate of 78% (SD = .06) and those with a 500MS delay a mean of 61% (SD = .21).

The curve of successful expert usage rate over blocks using Equation 4.1 is very similar to the results from experiment 1 (compare Figure 5.3 to Figure 4.5). Both delay times follow the same pattern, and a constant offset factor between them. Participants achieved asymptotic performance in this experiment as well using our hypothetical one-hundredth block (Table 5.1).

5.2.2 Recall Rate

Section 4.2.2 provides a description of the different recall measures. We found no significant differences in recall for any of the recall tests (Figure 4.6).

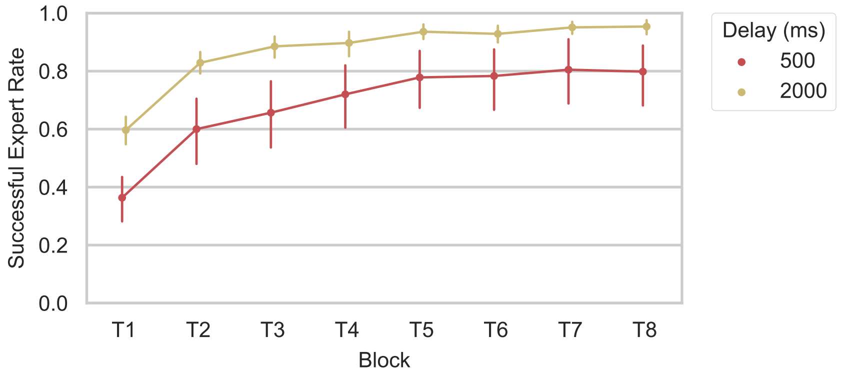


Figure 5.3: Successful expert rate for each DELAY by BLOCK. Blocks T1 to T8 occurred during the main session (see Figure 4.3 for session and block progression).

Instantaneous Recall — We found no effect of DELAY on instantaneous recall ($t_{-0.33} = 51$, $p = 0.7$).

Short Term Recall — We found no effect of DELAY on short term recall, data failed the test of equal variance and Welch's t-test was used ($t_{-1.59} = 30$, $p = 0.1$).

Long Term Retention — There was no effect of DELAY on long term retention ($t_{0.17} = 33$, $p = .9$). There was also no effect of confidence ($t_{-0.96} = 33$, $p = .34$).

Recall after Relearning — Following relearning there was also no effect of delay on recall rate ($t_{-0.26} = 33$, $p = 0.8$).

5.2.3 Number of Errors

The total number of trials with an error in a block follows a similar trend to the first experiment. Many errors near the beginning of the experiment, then they reduce over time (Figure 5.5). A t-test shows a significant effect of DELAY on errors ($t_{-2.29} = 44.97$, $p < 0.05$). Participants using the shorter 500MS delay made approximately 6 fewer errors than those in the 2000MS condition. Participants in the 500MS condition made 12 errors over the entire experiment on average, whereas those in the 2000MS condition made 18.75 errors.

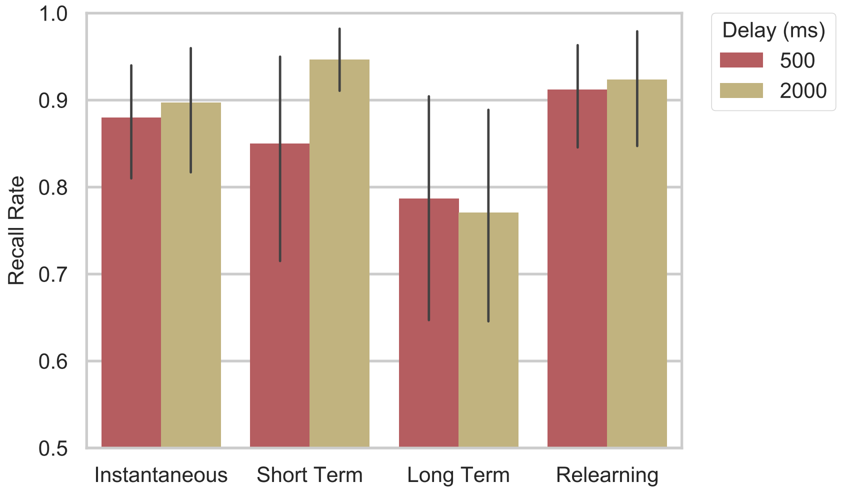


Figure 5.4: (Experiment 2) Recall rates at various measurement times. During the main session, *instantaneous* recall (R1) occurred directly following the training blocks and *short term* recall (R2) occurring after a distractor task. During the follow-up session 24 hours later, *long term* retention (R3) was measured upon entering the session and *relearning* (R5) after performing a single training block.

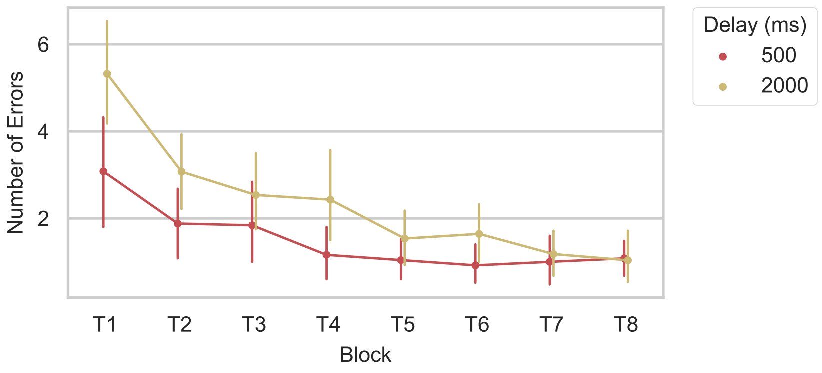


Figure 5.5: (Experiment 2) Number of trials containing an error by BLOCK and DELAY.

5.2.4 Selection Time

Unlike Experiment 1, we found no difference in selection time. We found no significant difference in selection time between the two conditions ($t_{1.26} = 35.78$, $p < 0.2$). The median

Table 5.2: Experiment 2: Mean and standard deviation of NASA-TLX results.

	Mental	Physical	Temporal	Performance	Effort	Frustration
500MS	4.9 (1.4)	3.5 (1.8)	4.7 (1.6)	5.8 (1.1)	5.5 (1.2)	3.0 (1.8)
2000MS	5.0 (1.2)	2.7 (1.7)	4.1 (1.5)	5.6 (0.9)	6.2 (1.0)	3.0 (1.8)

selection time for 500MS was 745ms (SD = 332) and the median selection time for 2000MS was 740ms (SD = 178).

5.2.5 Perceived Workload

Section 4.2.5 provides a description of the NASA-TLX questionnaire we used. As in the first experiment, there was no significant difference in *Frustration* and overall participants rated the experiment as having low frustration.

We found a significant difference in *Effort* ($Z = 225$, $p < .02$. Table 5.2 provided all NASA-TLX results. Participants in the 500MS condition rated *Effort* above neutral (5.48, SD = 1.23) and those in the 2000MS condition rated it well above neutral (6.18, SD = 1.02). All other differences were not significant ($p < .09$).

5.3 Discussion

These results show the main trends of the first experiment generalise to another style of rehearsal-based menu. Comparing the results of the two experiments, the rate of successful expert selections was improved by a similar magnitude for both Fast2Click and the marking menu. Participants made fewer errors using the Fast2Click menu, and as a result, completed more successful expert selections. Although the magnitude of change in expert selections was similar, Fast2Click users completed more successful expert selections than marking menu users.

Increased effort for longer delay.

Overall, participants using the 2000ms delay rated it as requiring more effort. The increase in effort might be due to a higher proportion of menu selections using expert mode. Learning expert mode requires more effort than relying on the novice guide, but it is also more difficult to make a selection with expert mode in Fast2Click.

We can model the task based on the number of parameters a user must remember. In a one-level marking menu the user must remember a single piece of information, the angle. However, Fast2Click requires the user to remember a two-dimensional x and y position relative to the menu activation button. This means that even after remembering items, executing them without any visual guidance is still more difficult than with a marking menu. We suspect that the increased rate of expert use for the longer delay lead to a significant difference in effort because of these factors. Some of these factors might have existed for the marking menu experiment, however users also experienced an improvement in selection time, which might have reduced their sensitivity towards increased effort.

A longer Fast2Click delay does not improve selection time.

Unlike with marking menus, the median selection time was not significantly different for a longer delay. We believe this is linked to the same factors as the increased effort for the longer delay. Executing an expert mode selection in Fast2Click can be difficult because there is no feedback or spatial guidance. In the marking menu, participants can see a mark as they draw, which may help to visually connect the action with the item direction. Adding borders or artificial landmarks [53] could be a useful improvement to Fast2Click.

This generalization for the effect of delay across two menu types is encouraging, but it did not explore the recall ceiling effect. The following experiment uses a 15-item Fast2Click menu to generalize further and explore an expanded recall ceiling.

Chapter 6

Experiment 3: Larger Menus

This experiment extends our results from the previous experiments to a menu with more items. It also adjusts the protocol in order to also examine of recall during training. This study was conducted on Amazon Mechanical Turk using the same apparatus as the previous experiments.

6.1 Experiment Description

6.1.1 Menu

Participants use a 15 item Fast2Click menu 6.1. We expanded the size of the menu simply by adding an additional column and row. By using a Fast2Click menu instead of a marking menu, we remove the need for introducing a hierarchy of menu levels.

6.1.2 Tasks

Participants use the Fast2Click menu for a training and test task, both of which were identical to the previous two experiments (Section 4.1.2). Due to the ineffectiveness of the distractor task, and to keep the experiment's duration to be similar to our previous experiments, we removed the distractor task.

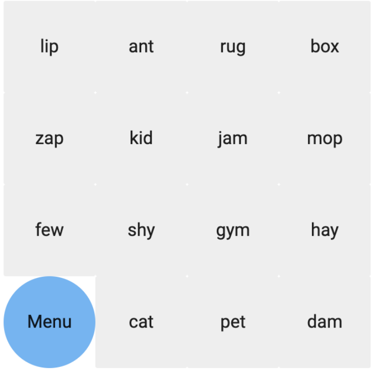


Figure 6.1: Fast2Click menu with 15 items. All other aspects of the menu remained the same.

6.1.3 Protocol

In this experiment we alter the main session's protocol in order to observe the impact of delay on recall rates over time (Figure 6.2). The main session begins and ends with the same questionnaires as in the previous experiments. Following the pre-questionnaire, participants complete two training blocks with 31 menu selections in each. After completing two consecutive training blocks they complete a test block for all 15 items. The alternating pattern of 2 training blocks and 1 test block is repeated 4 times for a total of 8 training blocks and 4 test blocks. The follow-up session remained identical to the previous two experiments (Section 4.1.3).

6.1.4 Design

The design was identical to the the second experiment (Section 5.1.4) except the Zipfian distribution was expanded to accomodate 15 items: 9, 4, 3, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1. The new words (dam, few, gym, hay, jam, kid, zap) were chosen such that no two words in the entire set start with the same letter.

6.1.5 Participants and Apparatus

We recruited 59 crowdworkers from the United States. Participants were paid $3 instead of $2 because the main session was 5 minutes longer. Participants were paid an additional

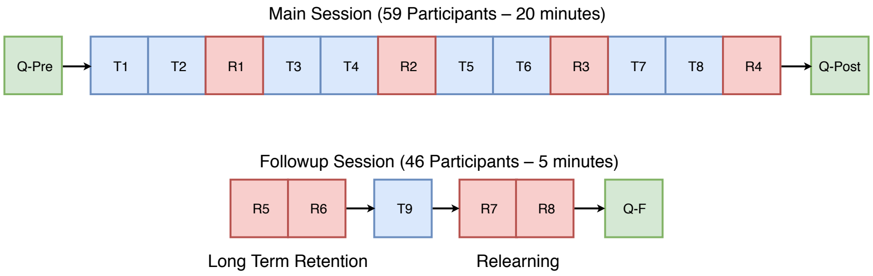

Main Session (59 Participants – 20 minutes)
Q-Pre
T1
T2
R1
T3
T4
R2
T5
T6
R3
T7
T8
R4
Q-Post
Followup Session (46 Participants – 5 minutes)
R5
R6
T9
R7
R8
Q-F
Long Term Retention
Relearning

Figure 6.2: Experimental procedure for experiment 3. Green blocks prefixed with Q indicate a questionnaire, blue blocks prefixed with T are training blocks, and red blocks prefixed with R are recall blocks. Between consecutive blocks a 5 second rest is enforced. Before beginning a section of the experiment the participants complete a short tutorial to familiarise them with the interface. Additional recall blocks were inserted during training test memorisation of commands over time.

$1 if they returned to the follow-up session. Only crowdworkers who did not participate in the previous two experiments were allowed to participate in this experiment.

6.2 Results

The mean age of 59 participants was 37.9 (SD = 11), 34 were female and 25 were male. 46 of the main session participants returned for the follow-up session (77%). There was 1 participant who used their left hand to control their pointing device. Participants used various pointing devices: 54 used a mouse, 4 used trackpads, and 1 participant used a trackball.

We removed a total of five outliers: two participants were removed for taking a screenshot of the menu; another two were removed for making more than 300 errors which was more than 3 standard deviations away from the mean; and one participant for only making a single expert selection which was more than 3 standard deviations away from the mean. After removing outliers, the 2000MS condition had 27 participants and 500MS had 28 participants.

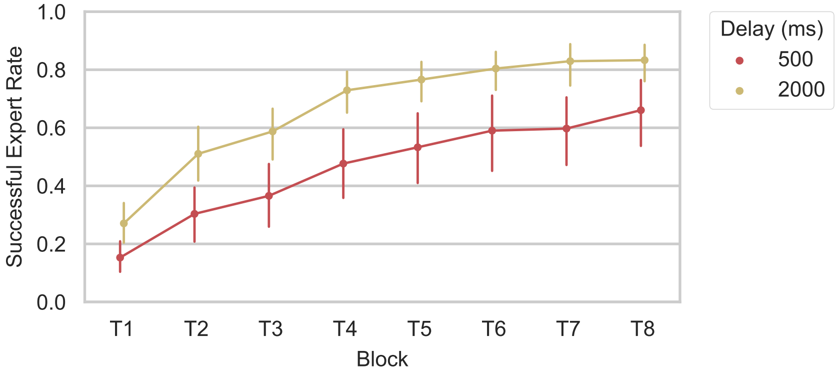


Figure 6.3: Experiment 3 successful expert rate for each DELAY by BLOCK.

Table 6.1: (Experiment 3) Exponential regressions of *expert use* by *block* for each DELAY using Equation 4.1. The y-intercept is calculated to aid interpretation, and $f(7)$ and $f(100)$ are calculated expert use rates for block 7 (i.e. block T8) and a hypothetical block 100 to show near-asymptotic characteristics.

Delay	R2	a	b	c	y-intercept	f(7)	f(100)
500MS	0.79	-0.61	0.24	0.76	0.16	0.65	0.76
2000MS	0.71	-0.59	0.45	0.87	0.28	0.84	0.87

6.2.1 Successful Expert Rate

We found an increase in expert usage rate of approximately 18% for the 2000MS condition compared to 500MS. This is again comparable to the increases between same conditions with 8-item Fast2Click and marking menus in the two previous experiments. Welch's t-test found a significant difference between 500MS and 2000MS ($t_{-3.41} = 44.8$, $p < 0.01$). The successful expert rate for 500MS was 0.41 (SD = 0.23) and for 2000 0.59 (SD = 0.15).

With more menu items the expert rate curve is more gradual, but we see a similar constant factor between the two delays as seen in the previous experiments. In addition to the main result we also can also see a more linear increase in successful expert use.

We were still able to fit the exponential curve function (Equation 4.1) to the data. The two different delays have near identical y-intercepts which suggests the 2000MS delay has a steeper gradient. Furthermore, the extrapolated 100th block ($f(100)$) is much higher than the final training block ($f(7)$), suggesting participants were still learning menu locations.

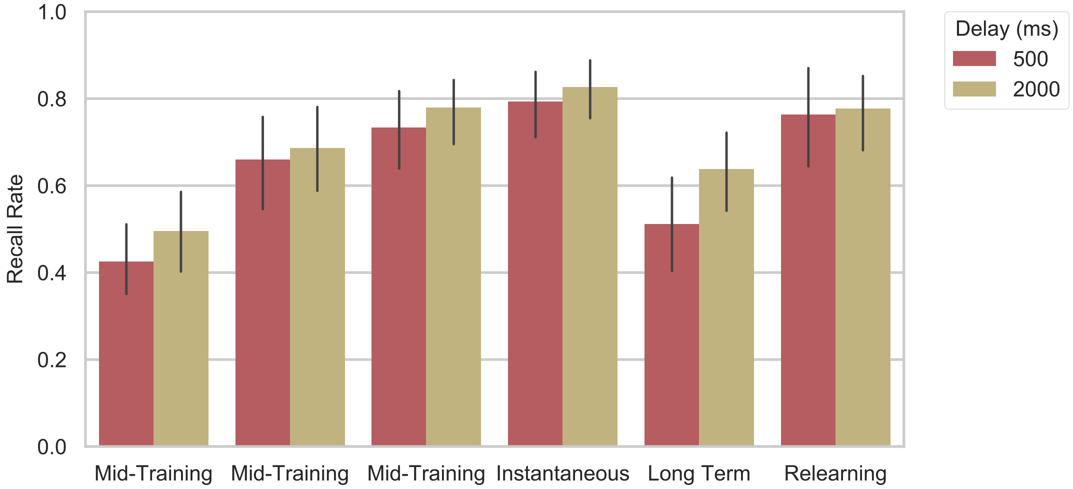


Figure 6.4: Experiment 3 Recall rates at key measurement times. During the main session, three mid-training recall tests (R1, R2, R3), then the *instantaneous* recall (R4) occurred directly following all the training blocks. During the follow-up session 24 hours later, *long term* retention (R5) was measured upon entering the session and *relearning* (R7) after performing a single training block.

6.2.2 Recall Rate

Section 4.2.2 provides a description of the instantaneous, long term retention, and after relearning recall measures as well as the method to measure confidence. Mid-training recall was measured after every two training blocks. Unlike the previous two experiments, all recall rates appear similar between delay conditions. There were no significant differences between 500MS and 2000MS for any recall measure: mid-training recall blocks (R1, R2, R3); instantaneous recall (R4); long term retention (R5); after relearning (R7); or the confidence measures (R5 vs. R6 or R7 vs. R8). Despite the lack of significant differences, there may be some evidence that participants in the 2000MS condition scored higher on some recall tests (Figure 6.4). Combined recall results of both 500MS and 2000MS for the first block began at 46% and increased to 81% by the end of the main session. During the followup session recall rates were 58% before the relearning blocks, and 77% afterwards.

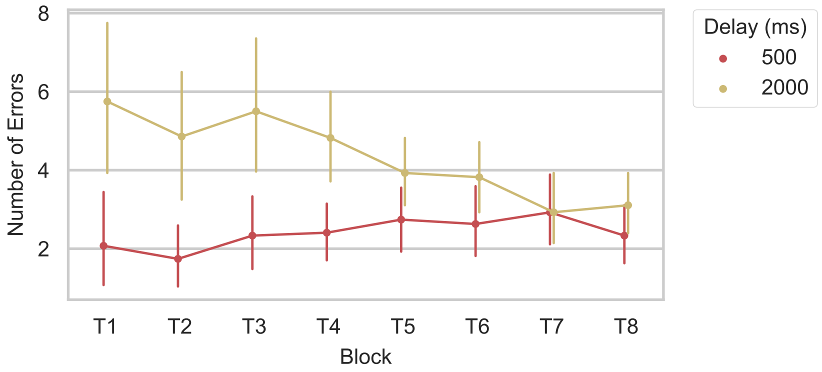


Figure 6.5: Experiment 3: Error results by block and condition.

6.2.3 Number of Errors

There were many more errors in the longer delay condition. The total number of errors aggregated across the entire experiment is significant between DELAY ($t_{-3.85} = 48.84$, $p < 0.001$). Participants using a 500MS delay have a mean of 19 errors (SD $= 12.3$) and those in the 2000MS condition a mean of 35 errors (SD $= 17.3$).

6.2.4 Selection Time

Like experiment 2, we found no difference in average selection time and the data has more variance. No significant main effect was found for DELAY on selection time ($t_{0.50874} = 41.96$, $p = .6$). Participants in the 500MS condition made selections with a mean time of 1245ms (SD $= 464$) and those in the 2000MS condition in 1152ms (SD $= 854$).

6.2.5 Perceived Workload

Section 4.2.5 provides a description of the NASA-TLX questionnaire we used. Results from the NASA-TLX data had a similar trend when compared to the previous two experiments (Table 6.2). When compared directly to experiment 2, the overall magnitude of responses was higher suggesting that performing command selections with a larger menu is more difficult when compared across all NASA-TLX measures. We found no significant differences between the two conditions for NASA-TLX scores. For subjective data, we used the Mann-Whitney test to analyse the data due to some data not being normally distributed.

Table 6.2: Experiment 3: Mean and standard deviation of NASA-TLX results.

	Mental	Physical	Temporal	Performance	Effort	Frustration
500MS	5.6 (1.0)	2.9 (1.5)	3.9 (1.4)	5.1 (1.4)	6.1 (0.8)	4.0 (1.7)
2000MS	6.1 (0.8)	3.3 (2.0)	3.5 (1.8)	5.1 (1.0)	6.3 (0.9)	3.7 (1.9)

6.3 Discussion

This experiment provides more evidence that a longer delay increases successful expert use. In addition, it also shows the general pattern of results when using 8-item menus (from the previous two experiments) generalises to a menu with more items.

Failure to find evidence that longer delays improve recall.

One goal of this experiment was to avoid a suspected ceiling effect because participants only had to remember 8 items in the first two experiments. With only eight items, the recall rates for delays above 200ms may not be significantly different because, with longer delays, participants were already recalling the majority of items. Since there were no significant effects of delay on recall using a 15-item menu, we now believe similar recall measures for longer delay times are not an artefact of a ceiling effect.

This suggests the difference in recall for very short delays, like those noted for 200ms in experiment 1, are due to very short delays actually harming recall. Said another way, there is something about delays shorter than some 200 to 333ms threshold that prevent people from “using” the delay to assist them in remembering menu items. Very short delays may not be functioning as a reactivator, instead they may be perceived as an activation lag in the menu interface. This would corroborate our earlier suggestion that the improvement in recall rate is due to longer delays being perceived as a purposeful part of an interface, not an artefact of a (poor) technical implementation of an interface.

Lack of increased effort.

Delay had no effect on any of the NASA-TLX measures, however in the previous experiment we detected increased effort for longer delays. We hypothesised that the effect in experiment 2 was due to the lack of motor guidance when performing an expert mode selection. The lack of an effect in this experiment supports our hypothesis because there is reduced overall expert use and therefore experienced the poor expert interaction fewer times.

Crowdworkers can remember many menu items.

Upon returning after 24 hours, the crowdworkers successfully remembered the locations of over half the menu items. This suggests that crowdworkers really gave the experiment a large portion of their attention and mental effort. Additionally, it further suggests that rehearsal-based interfaces are an effective method for improving user memory of menu locations.

Chapter 7

Conclusion

7.1 Design Implications

Delay is shown to have a profound effect on user performance and behaviour with rehearsal-based interfaces. Designers and researchers should take this into account when designing these interfaces.

Shorter delays should be avoided. An obvious implication of our results is that rehearsal-based interfaces should avoid short delays because they harm user's ability to learn command locations and slows down their command selections. We suspect users perceive very short delays as lag rather than a feature of the interaction technique. Users therefore fail to spend that time attempting to retrieve the menu location they are looking for. Additionally, if the delay is too short, and triggers before the user is able to initiate the motor action, it distracts from completing a selection resulting in a slower performance time. In general, we recommend delays less than 333ms should be avoided in rehearsal-based interfaces.

Using short delays may be appropriate if the primary goal is not expert efficiency and long term time-based performance, but rather to mitigate errors and improve the discoverability of the guidance interface. Yet, even in this case, our recommendation is that the delay be removed entirely rather than use a short delay like 200ms. Using a zero delay should retain the low error and high discoverability benefits of the short delays we tested, without harming recall or interrupting the motor action.

Longer delays should be adopted. We recommend that longer delays should be adopted for applications where users are interested in achieving long-term, efficient interaction.

Interface designers should consider using much longer delays than used previously since this will ead to more successful expert selections. In our experiment task, participants did not appear to be frustrated by a two-second delay, so this may be practical for real interfaces. Of course the direct application of our finding to real interfaces is more nuanced, and there may be practical reasons why such long delays are not desirable. Regardless, our main guideline is: "delay novice mode for at least 500 milliseconds, perhaps longer". In applications where errors have a high impact, delays of only 500ms should be used. However, for applications where errors have low impact, then longer delays such as one to two seconds will be beneficial.

The detrimental effect longer delays have on discoverablility could potentially be overcome with interaction techniques. For example, using a subtle timer display to indicate a long delay is in progress, or provide brief feedback indicating a novice mode is available upon initial menu activation. These kinds of techniques would signal to the user that the delay is deliberate rather than a result of lag.

Delays should be customisable. Results from other studies indicate that there is an all-or-nothing effect when using rehearsal-based interfaces [35]. Because some users may never adopt expert mode behaviour, it would be sensible to provide a user controlled customisation for delay length. Some examples of this approach already exist, such as in the Blender 3D modelling application which uses pie menus featuring customisable delay lengths. However, allowing users full customisation down to the millisecond might be confusing or even detract from the gains they could receive. We recommend delay customisation to be discrete rather than continuous, and different delay options should be presented as expected benefits. For example, providing options of no delay to reduce errors, a very long delay, like 1 or 2 seconds, to increase performance over the long term, and a delay time in the middle, such as 333 to 500ms, to balance these two trade-offs.

Interfaces with delays should be compared carefully in academic literature. Our results show how critical the delay is to user performance, and researchers should consider this when comparing rehearsal-based interfaces. For example, the results from the evaluation of Luo and Vogel's Pin-and-Cross menu [41] suggest that Pin-and-Cross was 291ms faster than a marking menu. However, the marking menu they compared to had a 333ms delay before activation and an additional 150ms delay before displaying guidance. This additional delay could have improved performance of the marking menu, suggesting that Pin-and-Cross's performance improvement is even greater than the evaluation suggests.

Delay should be considered for other interfaces. The benefits of longer delays in rehearsal-based interfaces suggests possible applications to other interfaces for other usage scenarios. For example, delays might also be useful to manipulate "satisficing" (see Section 2.1) by delaying the opening time of a linear menu to nudge users to consider keyboard shortcuts or a context menu instead. Even farther afield, is a use case related to problematic internet use [34]. Some users experience this when they habitually open a new tab and begin typing the address to a time wasting site. A long delay could be inserted just before the tab opens to provide a small moment for reflection, perhaps break the habit, and ultimately reinstate that user's control over their internet use. Of course, these examples are task-level behaviour rather than command-selection behaviour, and our menu-focused results might not extend to them. Further research is required to investigate how delays can be incorporated to these and other user interfaces.

7.2 Conclusions

Users often fail to transition from novice to expert, and rehearsal-based interfaces are designed to close the gap. However, many users do not adopt expert use. This thesis provides empirical evidence that using delay time as a rehearsal activation method is not only effective at encouraging this transition, but it becomes a tune-able parameter for the rate of expert use. Our experiments show that across two different menu styles and two item complexity levels, a very short delay can actually harm user recall, but a longer delay increases expert usage without introducing measurable frustration or increasing overall selection time. In fact, a longer delay time will likely reduce selection time in the long run.

7.2.1 Limitations

Like any controlled experiment, there are limitations resulting from scope and design choices that are necessary to make the protocol practical.

Our results do not necessarily extend to extremely large menus. Although more menu items are harder to remember, users do not need to use all of them because command selection follows a Zipfian distribution. In fact, users remember just about as many selections after a single block of training with an 8-item menu as they would after 8 blocks with the 15-item menu. Care does need to be taken when using more larger menus to ensure error rates do not rise above an acceptable level. The above can be mitigated using a rehearsal mediator as explained below in future work.

Our results do not consider any additional effects for intentional mappings between menu locations and items. Our experiment designs were constructed to make mappings between menu locations and items arbitrary. The set of three-letter words avoids introducing additional memory cues based on word length or meaning, and the random mapping avoids grouping similar items (like grouping commands for colour on one line of a FastTap menu), or spatial conventions (such as “prev” and “next” being mapped to left and right marking menu directions). This was necessary to avoid experimental confounds, but it also means that our results may be closer to a lower bound on how much expert use is increased. User performance with real systems, where mappings between menu locations and items can be carefully designed to maximise intuition and memory, may amplify or dampen the effect of longer dwell times.

Our results do not necessarily extend to even longer delays. At first blush, our results could be interpreted to suggest very, very long delays, or even infinite delays, would result in higher and higher rates of expert use. Of course, there would be a point where the delay time becomes so long that users may not discover novice mode, or if they were aware of novice mode, actually waiting to activate it would have such a high cost that no one would use it. We originally expected two seconds to be pushing the limits of user frustration, but as we show above, that was not the case. Searching for a practical maximum delay time may be of interest in the future, but this question is independent to the present results. We already show a general trend that longer delay times within a reasonable range are beneficial.

Our results do not necessarily extend to zero delay. We did not test a zero delay because then the interface has no reactivator, so there is no explicitly controlled expert mode and the interface is no longer rehearsal-based. Using a zero delay would also mean we could not empirically measure whether the user intended to view the menu item locations (as an implicit novice mode), or make a selection without consulting them (as an implicit expert mode). Recall could be measured, but given our results for the imperceptible, but controllable 200ms delay condition, we expect zero delay will also harm recall.

One other issue with zero delay is that the menu location visualisation could be distracting since it will momentarily flash on the screen with every selection. Kurtenbach et al. actually cited visual distraction as a primary justification for using some delay [32]. However, Guimbretière et al.’s FlowMenu always shows a semi-transparent visualisation [22], and the authors conjecture that a delay would have a negative temporal impact for novice users, and experts are not distracted by it. Neither paper formally evaluated distraction effects, and our results provide evidence that a temporal impact for novices is minimal. Regardless, evaluating a zero delay interface in terms of recall and visual distraction is an interesting avenue for future work.

Our results do not necessarily extend to tasks with high error costs. As discussed above, our experiment task models very low error cost. When an error occurred, the participant was notified and they were required to fix the error by selecting an item again before continuing. We believe this approximates error recovery using a low effort action, like pressing the "undo" shortcut key. However, there are cases in a real system where error recovery is more time consuming. Consider accidentally overwriting a file with "save" instead of "save as", and then needing to restore the file from a backup system. Experimentally manipulating error cost, such as adding a time or extra task penalty each time an error occurs, would enable a closer study of how perceived time difference affects expert use.

Our results do not necessarily extend to interfaces requiring motor learning. The two menu interfaces we tested use simple movements, a directional drag for the marking menu and a directional tap for Fast2Click. It is possible that the effect of delay may be different for rehearsal-based interfaces requiring motor learning, such as the gestural interface evaluated by Anderson and Bischof [1]. We see this as a natural direction for future work.

7.2.2 Future Work

Like any new result, our findings suggest more avenues of enquiry. For example, the duration of a delay may not necessarily need to be statically fixed. If delay were a function of the total number of menu selections it could support the user throughout their development from novice to expert. To improve discoverability initially, a small delay might work well. Then, to help the user learn item locations, the delay could slowly increase with time. A related kind of continuous adjustment of novice visualisation was shown to be effective for learning shape gestures [1].

Our descriptive model of rehearsal-based interfaces also provides a framework for exploring other types of reactivators. For example, pressing the command key instead of waiting for a delay might lead to better rehearsal benefits, because two different hands are used and the motor action of the hand executing the command is identical regardless of novice or expert modes.

Long delay times might also reduce the "all-or-nothing effect" noted by Lafreniere et al. [35]. A longitudinal study of longer delays might help improve the adoption rate for using expert mode in the long term, and provide those users with a long term performance efficiency benefit that they would miss otherwise.

We hope our results provide insights and design guidelines to make the seemingly benign choice of delay time more principled. Perhaps with longer delay times in all rehearsal-based interfaces, everyone may achieve more expert-level performance.

References

[1] Fraser Anderson and Walter F. Bischof. Learning and performance with gesture guides. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1109–1118, New York, NY, USA, 2013. ACM.

[2] Jonathan Back, Duncan P. Brumby, and Anna L. Cox. Locked-out: Investigating the effectiveness of system lockouts to reduce errors in routine tasks. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 3775–3780, New York, NY, USA, 2010. ACM.

[3] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. *Wave Menus: Improving the Novice Mode of Hierarchical Marking Menus*, pages 475–488. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[4] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 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, pages 15–22, New York, NY, USA, 2008. ACM.

[5] Roger Bakeman. Recommended effect size statistics for repeated measures designs. *Behavior research methods*, 37(3):379–384, 2005.

[6] Olivier Bau and Wendy E. Mackay. 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, pages 37–46, New York, NY, USA, 2008. ACM.

[7] William Buxton. A three-state model of graphical input. In *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction*, INTERACT '90, pages 449–456, Amsterdam, The Netherlands, The Netherlands, 1990. North-Holland Publishing Co.

[8] John M Carroll. *HCI models, theories, and frameworks: Toward a multidisciplinary science*. Elsevier, 2003.

[9] John M. Carroll and Mary Beth Rosson. Interfacing thought: Cognitive aspects of human-computer interaction. chapter Paradox of the Active User, pages 80–111. MIT Press, Cambridge, MA, USA, 1987.

[10] O. Chapuis, R. Blanch, and M Beaudouin-Lafon. Fitts law in the wild: A field study of aimed movements. Technical report, Universite de Paris Sud, Orsay, France, 2007.

[11] Andy Cockburn, Carl Gutwin, and Saul Greenberg. A predictive model of menu performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 627–636, New York, NY, USA, 2007. ACM.

[12] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. Supporting novice to expert transitions in user interfaces. *ACM Comput. Surv.*, 47(2):31:1–31:36, November 2014.

[13] Andy Cockburn, Per Ola Kristensson, Jason Alexander, and Shumin Zhai. Hard lessons: Effort-inducing interfaces benefit spatial learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 1571–1580, New York, NY, USA, 2007. ACM.

[14] S. R. Ellis and R. J. Hitchcock. The emergence of zipf's law: Spontaneous encoding optimization by users of a command language. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(3):423–427, May 1986.

[15] Leah Findlater and Joanna McGrenere. A comparison of static, adaptive, and adaptable menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 89–96, New York, NY, USA, 2004. ACM.

[16] Leah Findlater, Karyn Moffatt, Joanna McGrenere, and Jessica Dawson. Ephemeral adaptation: The use of gradual onset to improve menu selection performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1655–1664, New York, NY, USA, 2009. ACM.

[17] Paul Morris Fitts and joint author Posner, Michael I. *Human performance*. Belmont, Calif Brooks/Cole Pub. Co, 1967. Bibliography: p. 151-158.

[18] Bruno Fruchard, Eric Lecolinet, and Olivier Chapuis. Markpad: Augmenting touchpads for command selection. In *Proceedings of the 2017 CHI Conference on Human*

Factors in Computing Systems, CHI '17, pages 5630–5642, New York, NY, USA, 2017. ACM.

[19] Wai-Tat Fu and Wayne D. Gray. Resolving the paradox of the active user: stable suboptimal performance in interactive tasks. *Cognitive Science*, 28(6):901 – 935, 2004.

[20] Wayne D Gray and Deborah A Boehm-Davis. Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied*, 6(4):322, 2000.

[21] Tovi Grossman, Pierre Dragicevic, and Ravin Balakrishnan. Strategies for accelerating on-line learning of hotkeys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 1591–1600, New York, NY, USA, 2007. ACM.

[22] François Guimbretiére and Terry Winograd. Flowmenu: Combining command, text, and data entry. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, pages 213–216, New York, NY, USA, 2000. ACM.

[23] Carl Gutwin, Andy Cockburn, and Benjamin Lafreniere. Testing the rehearsal hypothesis with two fasttap interfaces. In *Proceedings of the 41st Graphics Interface Conference*, GI '15, pages 223–231, Toronto, Ont., Canada, Canada, 2015. Canadian Information Processing Society.

[24] Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. Faster command selection on tablets with fasttap. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2617–2626, New York, NY, USA, 2014. ACM.

[25] Sandra G. Hart and Lowell E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In Peter A. Hancock and Najmedin Meshkati, editors, *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139 – 183. North-Holland, 1988.

[26] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 453–456, New York, NY, USA, 2008. ACM.

[27] S. Komarov, K. Reinecke, and K. Z. Gajos. Crowdsourcing performance evaluations of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI 13, pages 207–216, 2013.

[28] Gordon Kurtenbach and William Buxton. 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, pages 482–487, New York, NY, USA, 1993. ACM.

[29] Gordon Kurtenbach and William Buxton. User learning and performance with marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 258–264, New York, NY, USA, 1994. ACM.

[30] Gordon P. Kurtenbach, Abigail J. Sellen, and William A. S. Buxton. An empirical evaluation of some articulatory and cognitive aspects of marking menus. *Hum.-Comput. Interact.*, 8(1):1–23, March 1993.

[31] Gordon Paul Kurtenbach. *The Design and Evaluation of Marking Menus*. PhD thesis, Toronto, Ont., Canada, Canada, 1993. UMI Order No. GAXNN-82896.

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

[33] Kurtenbach, Gordon. Demo of Marking Menus.

[34] Daria J Kuss and Olatz Lopez-Fernandez. Internet addiction and problematic Internet use: A systematic review of clinical research. *World journal of psychiatry*, 6(1):143–176, mar 2016.

[35] Benjamin Lafreniere, Carl Gutwin, and Andy Cockburn. Investigating the post-training persistence of expert interaction techniques. *ACM Trans. Comput.-Hum. Interact.*, 24(4):29:1–29:46, August 2017.

[36] Benjamin Lafreniere, Carl Gutwin, Andy Cockburn, and Tovi Grossman. Faster command selection on touchscreen watches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4663–4674, New York, NY, USA, 2016. ACM.

[37] David M. Lane, H. Albert Napier, S. Camille Peres, and Aniko Sandor. Hidden costs of graphical user interfaces: Failure to make the transition from menus and icon toolbars to keyboard shortcuts. *International Journal of Human Computer Interaction*, 18(2):133–144, 2005.

[38] Nathaniel Leibowitz, Barak Baum, Giora Enden, and Amir Karniel. The exponential learning equation as a function of successful trials results in sigmoid performance. *Journal of Mathematical Psychology*, 54(3):338 – 340, 2010.

[39] G. Julian Lepinski, Tovi Grossman, and George Fitzmaurice. The design and evaluation of multitouch marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2233–2242, New York, NY, USA, 2010. ACM.

[40] Wanyu Liu, Gilles Bailly, and Andrew Howes. Effects of frequency distribution on linear menu performance. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 1307–1312, New York, NY, USA, 2017. ACM.

[41] Yuexing Luo and Daniel Vogel. 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 &#38; Technology*, UIST '15, pages 323–332, New York, NY, USA, 2015. ACM.

[42] Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. Promoting hotkey use through rehearsal with exposehk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 573–582, New York, NY, USA, 2013. ACM.

[43] Sylvain Malacria, Joey Scarr, Andy Cockburn, Carl Gutwin, and Tovi Grossman. Skillometers: Reflective widgets that motivate and help users to improve performance. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 321–330, New York, NY, USA, 2013. ACM.

[44] Stephen Olejnik and James Algina. Generalized eta and omega squared statistics: measures of effect size for some common research designs. *Psychological Methods*, 8(4):434–447, December 2003.

[45] Stuart Pook, Eric Lecolinet, Guy Vaysseix, and Emmanuel Barillot. Control menus: Excecution and control in a single interactor. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '00, pages 263–264, New York, NY, USA, 2000. ACM.

[46] Joey Scarr, Andy Cockburn, Carl Gutwin, and Sylvain Malacria. Testing the robustness and performance of spatially consistent interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3139–3148, New York, NY, USA, 2013. ACM.

[47] Joey Scarr, Andy Cockburn, Carl Gutwin, and Philip Quinn. Dips and ceilings: Understanding and supporting transitions to expertise in user interfaces. In *Proceedings*

of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11, pages 2741–2750, New York, NY, USA, 2011. ACM.

[48] R. A. Schmidt, D. E. Young, S. Swinnen, and D. C. Shapiro. Summary knowledge of results for skill acquisition: support for the guidance hypothesis. *J Exp Psychol Learn Mem Cogn*, 15(2):352–359, Mar 1989.

[49] Katherine Schramm, Carl Gutwin, and Andy Cockburn. Supporting transitions to expertise in hidden toolbars. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4687–4698, New York, NY, USA, 2016. ACM.

[50] Susanne Tak, Piet Westendorp, and Iris Rooij. Satisficing and the use of keyboard shortcuts: Being good enough is enough? 25:404–416, 08 2013.

[51] Mark A. Tapia and Gordon Kurtenbach. Some design refinements and principles on the appearance and behavior of marking menus. In *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology*, UIST '95, pages 189–195, New York, NY, USA, 1995. ACM.

[52] Sherman W Tyler, Paula T Hertel, Marvin C McCallum, and Henry C Ellis. Cognitive effort and memory. *Journal of Experimental Psychology: Human Learning and Memory*, 5(6):607, 1979.

[53] Md. Sami Uddin, Carl Gutwin, and Andy Cockburn. The effects of artificial landmarks on learning and performance in spatial-memory interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3843–3855, New York, NY, USA, 2017. ACM.

[54] Md. Sami Uddin, Carl Gutwin, and Benjamin Lafreniere. Handmark menus: Rapid command selection and large command sets on multi-touch displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5836–5848, New York, NY, USA, 2016. ACM.

[55] Charles G Willis, Edith Law, Alex C Williams, Brian F Franzone, Rebecca Bernardos, Lian Bruno, Claire Hopkins, Christian Schorn, Ella Weber, Daniel S Park, et al. Crowdcurio: an online crowdsourcing platform to facilitate climate change studies using herbarium specimens. *New Phytologist*, 2017.

[56] Jingjie Zheng, Blaine Lewis, Jeff Avery, and Daniel Vogel. Fingerarc and fingerchord: Supporting novice to expert transitions with guided finger-aware shortcuts. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, pages 347–363, New York, NY, USA, 2018. ACM.