Revisiting Menu Design Through the Lens of Implicit Statistical Learning

Emmanouil Giannisakis
EPAM Systems
Berlin, Germany
em.giannisakis@gmail.com

Evanthia Dimara
Utrecht University
Utrecth, Netherlands
evanthia.dimara@gmail.com

Annabelle Goujon
Université de Bourgogne Franche-Comté
Besançon, France
annabelle.goujon@univ-fcomte.fr

Gilles Bailly
Sorbonne Université
Paris, France
gilles.bailly@sorbonne-universite.fr

ABSTRACT

Implicit Statistical Learning (ISL) studies how exposing individuals to repeated statistical patterns can help develop skills in the absence of conscious awareness, such as learning a language or detecting familiar shapes. This paper transposes ISL in the context of *menu design* learnability. Our analysis of menu patterns in various applications from the 80s to today reveals a consistent linear pattern with *command names* on the left and *keyboard shortcut cues* aligned on the right. We then develop a design space of menu patterns by manipulating two factors of ISL theory, spatial proximity (distance) and relative positioning between commands and shortcut cues. We empirically compare four menu patterns of this design space on whether they can improve keyboard shortcut adoption through two controlled experiments. Results did not capture clear effects among the menu patterns, suggesting that ISL in the context of HCI might involve more complex factors than initially anticipated, such as the time the users are exposed to the menu pattern. We reflect on the challenges in applying theories from cognitive science to HCI and hope that our systematic methodology and experiment designs will serve as a basis for encouraging more studies in the area.

CCS CONCEPTS

• **Human-centered computing**; • **Human-Computer Interaction (HCI)**; • **HCI design and evaluation methods**;

KEYWORDS

Implicit Statistical Learning, spatial relationships, GUI, menu

ACM Reference Format:
Emmanouil Giannisakis, Evanthia Dimara, Annabelle Goujon, and Gilles Bailly. 2022. Revisiting Menu Design Through the Lens of Implicit Statistical Learning. In *Proceedings of the 2022 International Conference on Advanced Visual Interfaces (AVI 2022), June 6–10, 2022, Frascati, Rome, Italy*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3531073.3531113

AVI 2022, June 6–10, 2022, Frascati, Rome, Italy

ACM ISBN 978-1-4503-9719-3/22/06…$15.00
https://doi.org/10.1145/3531073.3531113

1 INTRODUCTION

One of the main HCI challenges is to help users learn how to use an interface with as little effort as possible [33]. More effortful approaches such as manuals and tutorials are useful, but not always feasible in practice [18]. In contrast, designers and researchers strive to offer effortless and intuitive solutions [25, 32, 58], for example, by optimizing design layouts attracting user attention to relevant items and visual cues. Yet, HCI has limited comprehensive theories and frameworks to help us investigate systematically how users learn by being exposed to a repeated pattern.

An emerging field in cognitive science that studies how individuals learn new information by interacting with the environment is the Implicit Statistical Learning (ISL)[16, 19]. ISL investigates learning (or skill development) in the absence of conscious awareness. It explains how the brain discovers and encodes patterns (also called statistical regularities) within its repeated exposure to environment stimuli. To the best of our knowledge, ISL has not yet been investigated in the HCI literature. Typical ISL applications include how children learn a language effortlessly by listening to other people speak [38], or detection of visual shapes among a number of distractors [31]. Still, understanding and capturing underline processes of skill acquisition and visual search [18] can be of tremendous importance for effective user interface design.

Yet transposing ISL in the context of interface learnability is not straightforward given that individuals have difficulty reflecting on unconscious processes [49]. The aforementioned tasks and stimuli used in cognitive science experiments are typically low-level, carefully selected for measuring and understanding behavior. HCI experiments also aim to provide insights on how to design interactive systems or displays, while accounting for additional factors such as aesthetics, usability and functionality. In this paper, we explore a way of transposing the ISL theory in the context of HCI.

In particular, we focused on *menu design* learnability through the lens of ISL. At first, we operationalized the notion of statistical regularity exploring design menu conventions in approximately 200 menus in various applications from the 80s to today. Our analysis showed that menus tend to follow a consistent pattern of linear fashion which displays command names on the left and *keyboard shortcut cues* aligned on the right. Optimizing the ISL factors of *spatial proximity* and *relative positioning* [31], we developed a design space of alternative menu patterns. Both those ISL factors affect the visual arrangement of elements in GUIs which in turn is known to influence decision making processes of users [13] and reduce cognitive load [35]. We then consulted designer experts [43] to review our design space and identify promising menu patterns including criteria of aesthetics and readability. We evaluated the menu designs in two controlled experiments investigating the effect of our designs on *keyboard shortcut adoption*, a well-established HCI problem [32, 44] on how users can become more efficient by transitioning from menus to expert methods for command selection.

Results did not capture clear effects on which menu patterns are better for keyboard shortcut adoption, at least not within the time scale of our 1-hour lab experiment. While we remain positive on the importance of studying ISL pattern exposure in HCI, we reflect on a number of barriers along the process. We suggest that ISL in the context of HCI might involve more complex factors than initially anticipated, such as the time the users are exposed to the pattern and we hope that our systematic methodology and experiment designs will serve as a basis for more studies in this area.

2 BACKGROUND

In this section we first discuss the main principles of *implicit statistical learning* that we consider as relevant to UI design. We then review current design interventions to promote awareness and explicit learning in modern GUI.

2.1 Implicit Statistical Learning

Implicit Statistical Learning (**ISL**) [16, 19] integrates two contemporary approaches: *implicit* and *statistical* learning. Implicit learning [49] usually defined as the process of individuals acquiring knowledge without having intention to do so or being necessarily aware of it. The most common example is the early child language acquisition, i.e. how a child learns a language unintentionally without having a formal education [38]. This process yields abstract knowledge about the environment that individuals interact with [49]. Statistical learning refers to the unconscious process in which repeated patterns, or statistical regularities (e.g., probabilistic regularities of the environment that predict future events), are extracted from sensory inputs [59]. These two approaches are often published in separate literatures and sometimes interpret their data in a different way [16]. However, they describe the same phenomenon and provide similar results (for an overview see [47]). Consequently, several authors use the joint term *implicit statistical learning* to cover both approaches [16, 19].

2.1.1 Why Does ISL Occur? One of the explanatory theories that have been offered for ISL is the formation of chunks [28, 47]. Chunking characterizes the associative processing by which people bind together co-occurring elements or information from their interaction with the environment. Attention plays a critical role in the formation of cognitive units: perceptual primitives would only be grouped together to form a chunk when they are simultaneously held in a spatial-attention window, which is constrained by working-memory limitations [48].

One thing that remains unclear is the role of awareness in ISL. There has been some debates on whether learning without awareness can occur at all [23, 38] or what type of awareness can facilitate ISL. A recent study [38] for example focused on two types of awareness: (a) at the level of noticing and (b) at the level of understanding. An example of the former is when an individual learns a new language to notice that some words can take the suffix '-s' e.g. dogs and cats while an example for the latter is understanding that suffix '-s' signals plurality. However, prior work has shown that users tend to notice faster objects or entities that they are already familiar with, which could affect the ISL processes [53].

2.1.2 ISL on Skill Acquisition. Implicit statistical learning yields interesting results regarding the skill acquisition. First, ISL appears within multi-tasking, meaning that people have the ability to implicitly learn one task while executing another task [50, 59].This is especially interesting for our context, because we expect users to implicitly learn semantically or visually close elements while interacting with the interface. Second, ISL can help learners to reach faster automatization of performance without going through the initial cognitive demanding learning stage [37].

2.1.3 ISL on Visual Search. A form of implicit statistical learning is the contextual cueing [17]. Contextual cueing explains how contextual regularities present in the display can be implicitly detected and learnt during the visual search, optimising basic visual processing [31]. ISL appears sensitive to several factors [26, 31]:

- *Spatial proximity* is one of the key factors of contextual cueing. Although ISL on non-spatially close elements is possible, it occurs under far more restrictive conditions than those required for learning the relations between spatially close events [31, 47]. This phenomenon relates to the Gestalt's *law of proximity*, in which to perceive an assortment of objects, an individual forms as a group the ones that are close to each other. This law is often used in advertising logos to emphasize which aspects of events are associated.
- *Relative Spatial positioning*: recent studies [24, 61] highlight that relative positioning and spatial proximity play an important role on how individuals detect and learn contextual regularities.
- *Temporal proximity*: a delay of just 3s between two statistically contingent elements was sufficient to deteriorate intertrial learning in a contextual cueing task [56].
- *Accumulation* of instances in memory: only limited contingencies can be learnt in a restricted period, suggesting that only few associations trigger learning [55].

The aforementioned factors suggest potential ways to encourage the ISL of the UI while the user interacts with it. While performing a visual search in a User Interface, users are repeatedly exposed to several graphical elements (e.g. icon, label, text, etc.). It seems plausible that their spatial proximity, exposure timings, as well as the number of those elements can aid, or impede, the implicit learning of contextual information. Here we study the spatial proximity of the graphical elements, as it appears to be a key factor of ISL.

2.2 Skills, Attention & Awareness in HCI

Implicit learning is related to several, but different, phenomena such as sequence learning [34], visual search [30, 31, 56], attentional guidance [11, 57], cue-category association [51], causal learning [29] and motor learning [37, 46]. These phenomena have been extensively studied in HCI. For instance solutions such as ephemeral adaptation [25] or changing the background color of an element [58] attract the attention of the users towards this element to improve visual search in menus. Marking menus[40] favor skill acquisition by letting users repeatedly perform the same gesture.

2.2.1 *Experimental studies of ISL in Cognitive Science.*

ISL has been studied under several contexts and for each context the experimental protocols differ. The most relevant context for our use case is ISL in visual search which follows a standard experimental task [56]. During this task, participants search for a T-target within a configuration of L-distractors. Half configurations are systematically repeated across many blocks of trials. The others are presented only once during the task. A benefit on search times is typically observed in the repeated contexts compared to the novel contexts. This indicates that participants encode *implicitly* some elements of the context.

Related to our use case, Grossman et al. [32] compare several of these strategies to favor keyboard shortcut adoption. Interestingly, authors mentioned that these strategies favor "implicit learning" in opposition with "explicit learning". However, based on the ISL literature, these strategies primarily focus on explicit learning, not implicit learning, as they motivate users to **intentionally** learn keyboard shortcut. In contrast, we investigate whether repeatedly interacting with a graphical widget can help users to learn information in the surroundings without having the intention to learn them. Bailly, et al. proposed a theoretical model of shortcut adoption [9]. The model combines several cognitive mechanisms including implicit and explicit learning. Model fitting and model simulation suggest that implicit learning plays a role in shortcut adoption.

3 REVISITING MENU DESIGN

Our literature review on Implicit Statistical Learning (ISL) suggested that certain statistical regularities, or repeated patterns, can aid, or, conversely, impede implicit learning. This section attempts to identify and potentially improve such patterns in menu layouts.

3.1 Extracting Traditional Linear Menu Patterns

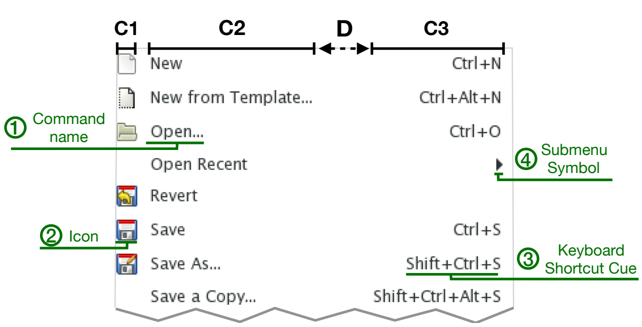

Figure 1: An example of the traditional linear menu of the inkscape application. The shortcut cue is placed far from the command name and on its right.

To extract menu design conventions, we analysed approximately 200 menus in various operating systems (Win, Mac, Linux) and tool-kits (e.g., Qt, Swing) from the 80s to today. Menus followed a linear, rectangular fashion divided into rows and columns. Each row is a menu item. The number of rows equals the number of available commands (plus the separators in semantic organizations). A typical linear menu organizes the elements in 3 columns (Figure 1):

-1st column contains command icons, the pictorial representations of functionalities (e.g., ,). Such icons depend on the application and the operating system, e.g., most Mac menus (and few Linux) do not display icons by default. The icon is sometimes replaced by a symbol ✓ or a widget ◉, ☑. Most menus display icons only for the frequent commands. Since all icons have the same size, their *alignment* is always fixed.

-2nd column contains *command names*, the textual labels representing the name of functionalities (e.g., Save, Copy). Ellipses can be added [5?](e.g. Save As...), if the command requires parameters (i.e. open a dialog box). Few menus (e.g., history menu in Safari) use an icon (e.g., website logo) in the command name. The *alignment* of the command names is always on the left.

-3rd column contains *keyboard shortcut cues* and submenu symbols (▶). The cues represent the sequence of keys to activate a functionality. Depending on the operating system the cue can be textual (e.g. Ctrl+S) or, as in Mac systems, a combination of symbol and text (⌘S). The symbol ▶, which does not have a cue, indicates hierarchical menu items (i.e. the item opens a sub-menu). The *alignment* of the cues is always on the right.

We saw very few exceptions of the above standards. An old Win 3.1 had shortcuts cues with left-alignment, Win had a 4rth column with ▶, and Blender had both cues and submenus.

Our analysis showed that the linear menu follows a consistent design pattern across operating systems and applications. The commands appear on the left, while the shortcut cues on the right. Similarly, the commands follow left alignment, while the shortcut cues follow right alignment. This standard introduces implications for the width of the columns. While the icons' width (1st column) is fixed, the command names (2nd column) depend on the longest. Similarly, the shortcuts' width (3rd column) depend on the shortcuts with the largest number of modifiers (max=3). Consequently, while the (relative) *spatial positioning* between the shortcut cue and the command is always the same (i.e. shortcut cue on the right), their *spatial proximity* varies a lot from one item to another and from one menu to another (see Figure 1). ISL theory suggests that such distance might impair the *implicit* learning of the keyboard shortcuts. Current literature further highlights that users often ignore keyboard shortcuts and they favor menus to select commands [6, 18, 32]. Existing solutions aim to increase users' awareness [32, 39, 44, 52], attract attention [32], inform about the relative performance [44] and/or change the incentives [52] for using the shortcuts. These approaches mainly focus on *when* and *how* to display information related to keyboard shortcuts, but they haven't investigated how the spatial relationship between the command name and the keyboard shortcut may affect the keyboard shortcut adoption.

3.2 Strategies to Improve the Menu Pattern

We identify several approaches favoring spatial proximity between commands and keyboard shortcut cues:

- *Decrease variability of command length (2nd column)*: As the width of the 2nd column depends on the longest command name, a designer could maintain similar command lengths. However, changing the wording of commands is a challenging task [7, 10]. Users are familiar with specific names, e.g., Save, Open while alternative command names do not always exist.

- *Decrease variability of shortcut cue length (3RD COLUMN)*: Similarly to the command names, the shortcut cues can also maintain the same width by reducing modifiers or replacing modifiers with symbols (e.g., Apple menus). Designers tend to anyways follow this solution, favoring simple shortcuts or using symbols. However, given the variability of the command names, this solution saves only a couple of pixels (as well as solutions that reduce padding or margins).
- *Change column alignment (2ND & 3RD COLUMN)*: This approach changes the position of the elements *within* columns. While the 2ND column is left aligned and the 3RD is right aligned several alternatives are possible while maintaining the column order.
- *Change position (2ND & 3RD COLUMN)*: This approach focuses on the relative position between the command name and the shortcut cue, typically, changing the order of the columns.

We focus on the two last strategies which do not modify the semantic of the command name or the shortcut cue.

3.3 Design Space: Position & Alignment

The design space of the linear menu is enormous involving numerous variables (e.g., font, saliency, icon design etc.) [10, 27]. We focus on the *position* and the *alignment* between the keyboard shortcut cues and the name of the commands.

To describe this design space, we use the iconified notation displayed in Figure 2. The grids on the left of each menu offer a quick overview of the set of possibilities and facilitate prototyping and brainstorming among designers. The command name is represented by the icon cmd and the keyboard shortcut cue by the icon ks. Each element is displayed in a box icon representing the whole space allocated for the elements, typically, the column width. The relative position (e.g., top, bottom, left, right) of the box represents the dimension *position*, while their relative position within the box represents their *alignment* (e.g., left , right, center). Figure 2 illustrates 8 menu instances of this design space by manipulating the position and the alignment of the two elements. A key feature of the design space is that the two elements can share the same box as shown in Figure 2.e.f. In that case, the designer is choosing to allocate one column for both elements. Consequently, this pair of elements will be following a unique alignment.

All combinations of positions and alignments between keyboard shortcut cues and command names derived 42 different menu patterns [1] More precisely, we considered 4 relative *positions* of the keyboard shortcut cue in relation to the command name: *left*, *right*, *below* or *above* (e.g., Fig2.c.d.h.g respectively). We also considered 3 *alignments* for each of these two elements: *left*, *right*, or *center* (e.g., for cues Fig2.b.c.d respectively). We also considered all (6) the menus where the elements are appended to each other. Given the configuration of these menus, we considered only 2 relative positions *left* or at the *right* (e.g., Fig2.f.e) and three alignments *left* (e.g., Fig2.e), *right*, or *center* (e.g., Fig2.f).

[1] Figure 2 shows only a subset of the menus. All menus are available in the supplementary material. See the osf link in Experiment 1.

3.4 Menu Pattern Selection

From the 42 menus, we discarded the 27 which were strongly sensitive to the length of the commands (e.g., in patterns .b and .d small command names are far from the shortcut cues). This systematic analysis resulted in *14 menu candidates* that increase the spatial proximity between the shortcut cue and the command name [2].

Yet, considering solely spatial proximity might affect other criteria of menu pattern design such as: *menu readability*, if commands and shortcut cues are easy to read; *menu aesthetics*,if they are aesthetically pleasing; and *frequency in UI*, if it is familiar in element organization in digital or physical documents (e.g. , applications, web pages, books, documents, journals). To account for those criteria, we consulted 4 design experts (28-48 years, 2 female, 1 male, 1 non-binary) of years of experience 3, 6-10 years and 1 > 10 years. On a 7-likert scale, the designers volunteered to evaluate the 14 menu patterns as well as the traditional linear menu pattern based on readability, aesthetics and frequency as well as the *shortcut noticeability* itself (i.e. if they think that the user will notice the keyboard shortcut of each command in this menu) Examples of the stimuli are the gray menus Figure 2.

Figure 3 reports the designers' responses. Each column corresponds to a menu pattern and the blue boxes the average rating for each criterion. Stronger blue indicates higher rating. Each menu can be identified uniquely by its position: cue on the (R)ight, (L)eft,(B)elow, (A)bove of the command and alignment: (R)ight, (L)eft, (C)enter, shown in the 2 last rows. Interestingly, the LEFT (Figure 2.c) was recognized as one of the most promising candidate, especially regarding shortcut noticeability.

We then performed a trade-off analysis to select the most promising menu patterns. On the one side, we considered the ratings of the designers for each menu. The relative importance of the criteria was: noticeability of the shortcut > > readability > > aesthetics > frequency. On the other side, we wanted to ensure that the selected menu patterns were different enough in terms of spatial relationships. Three promising menu patterns were retained which we will empirically evaluate next:

- the LEFT (Figure 2.c) where the keyboard shortcut cues are positioned before the command names.
- the RIGHT (Figure 2.e) where the keyboard shortcut cues are appended right after the command names.
- the BELOW (Figure 2.h) where the keyboard shortcut cues are located below the command names.

4 EXPERIMENT 1

Through the exploration of the design space we manage to identify three promising menu layouts however we don't know whether they will affect the keyboard shortcut adoption. Therefore, we compare the three aforementioned menu patterns to the traditional linear menu (BASELINE) on keyboard shortcut adoption. We also manipulate the LENGTH of the commands as it can influence spatial proximity (but not spatial positioning). Experimental material is available here: https://osf.io/sgqrf/?view_only=85faa79f72bc40119e380ab0030a53fe

[2] see osf.io link in Experiment 1 to access the full list of menu designs

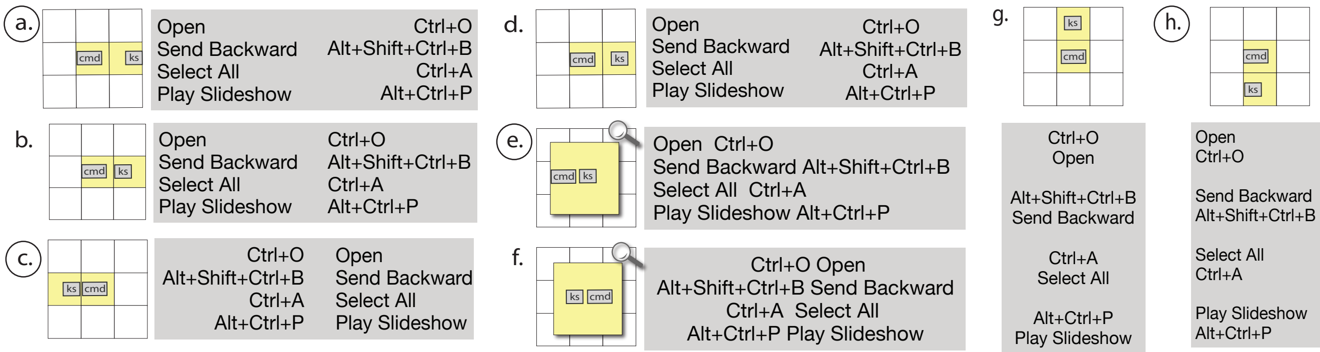

Figure 2: Design space for placing the keyboard shortcut cue (ks) in relation to its command name (cmd) regarding two dimensions: *position* and *alignment*. Shortcut positions: left (c,f), right (a,b,d,e) bottom (h), up (g) or diagonal (not shown in the figure). Shortcut alignments: left (b,h), right (a,c), and centered (f,g). The circled letters highlight the designs that we finally compared in Experiment 1.

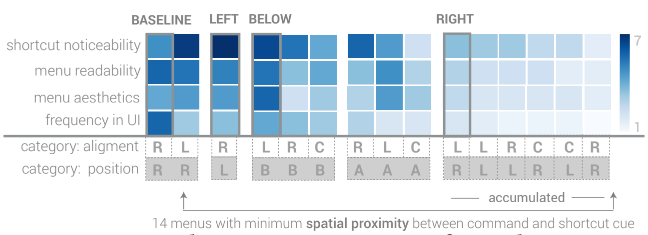

Figure 3: Expert designer average scores for each menu pattern.

4.1 Experimental design

Participants & Apparatus: We hired 27 university students (20-30 years) [3]. The payment was 10 €/hour and an extra bonus of 20, 15 and 10 € for the three fastest participants. We used a 23" screen, a keyboard and a mouse. The duration was 1 hour.

Menu, Targets & Frequency: This experiment follows the rational of the first Grossman's et al. study [32]: on the top of the screen, a menu system contained 4 dropdown menus of 12 items each. Each menu had a different layout [32]: LEFT, BELOW, RIGHT and the traditional menu layout (BASELINE) [4]. To compare both technique and individual item performance, we used an uniform frequency distribution (all items have the same frequency)[32]. The mapping of command names to keyboard shortcut cues followed the *"Bad"* quality rule of Grossman's et al. [32], i.e. the hotkey is not part of the command name. The design differed from the original only on command lengths, names and target lengths:

–*Command name lengths.* To create menus containing *"realistic"* variety of command name lengths and manipulate this factor, we used a database of about 30 frequent Mac OS applications (1048 menus in total) [8] and we computed the mean command name length (mean= 10 characters) and the variance [15] (24, with a 95% bootstrap confidence interval of [22, 25]). From this database, we randomly pick one 12-item menu with command name length distribution: 4-5-6-7-12-13-13-13-14-15-17-18. The mapping command name length-location was randomized for each menu.

–*Command names.* To fit the command lengths above (3 - 10 chars), we used only a subset of Grossman's categories [32]: *"animals"*, *"vegetables"*, *"fruits"* and *"office"*, synthesising command names with common expressions of 2-3 words (e.g., *"red fish"*) to precisely fit their length (i.e. number of chars).

–*Targets.* To use the command name length as a factor, each of the 4 menus contained 3 targets: the SMALL word (4 characters), the MEDIUM word (13 characters), and the LONG word (18 characters). This resulted in 12 target items in total (instead of 14 in [32]).

Stimulus & Task: On top of the screen, a menubar contained 4 buttons for each menu. The stimulus, displayed at the bottom, was an image depicting the target command (e.g. a *"red fish"*). The task was to select the target item as fast and accurately as possible using either the dropdown menu or the corresponding keyboard shortcut [32]. For wrong selections, a pop-up window appeared at the center of the screen. As a penalty, participants had to close this window and redo the task ([32] used 3 sec delay penalty instead). The next trial starts when participants executed the command correctly.

Procedure: Participants first performed a pre-test to ensure that they know what is a keyboard shortcut and how to execute it. We then explained participants the task encouraging them to be as fast and accurate as possible, emphasising also that they are free to use any method they want. We further mentioned that some previous studies indicate that using keyboard shortcuts can be faster. We also indicated that it is acceptable to make some mistakes in the beginning, to motivate risk-averse persons to use keyboard shortcuts. After 5 block of trials (each block consisted of 12 trials) a short dialog box appeared allowing the participants to take a 20" break. We then asked participants to fill out a post-questionnaire with gender, age and, how often they used keyboard shortcuts and perform a recall test where we showed them the visual stimuli they encountered during the session and asked them to indicate the corresponding keyboard shortcut.

Design: The experiment had two within factors: menu layout (BASELINE vs. RIGHT vs. BELOW vs. LEFT) and command length

[3]The planned size was 28, but one participant's data was lost due to a technical crush.

[4] Two main options exist for within design: either show the 4 conditions within the same menu system (ours and Grossman's [32]) or sequentially, one after the other. The drawback of the first option is that one layout motivating the users to use the shortcuts, might also motivates all conditions, resulting in observing smaller differences between conditions. However, the sequential option can also motivate the users to use shortcuts when switching from one condition to another, while more trials and mappings can overload participants' working memory [32].

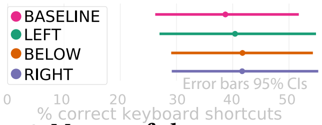

Figure 4: Mean% of shortcuts per menu.

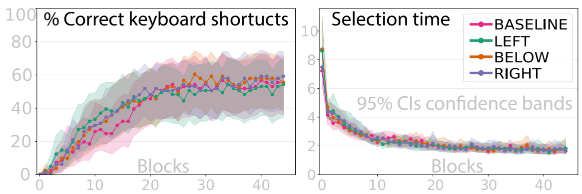

Figure 5: Mean% of shortcuts & time per menu

Figure 6: Mean% of shortcuts per command length

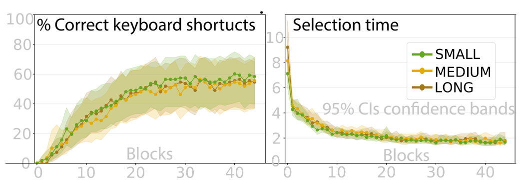

Figure 7: Mean% of shortcuts & time per command length

(SMALL, MEDIUM and LONG). The mapping was counter-balanced using a Latin square design. Participants performed 45 blocks of 12 selections. Each selection corresponds to one condition (Menu layout × command length). Selection order was randomized within block. Overall, the design was 27 participants × 45 blocks × 12 trials (3 command lengths × 4 targets) = 14580 selections.

Data analysis: We measured shortcut use as the mean rate of *correct keyboard shortcuts*, i.e the proportion of trials where participants successfully used the shortcuts without help. This dependent variable is often used to compare different designs promoting shortcut usage [4, 32, 44, 45] as it captures not only the fact that users use keyboard shortcuts but also correct mapping. Following recent criticism on p-values [22], we report and interpret our inferential statistics using bootstrapped confidence intervals (CI) [14].

4.2 Results

Errors bars in Figures 4,6,8 and confidence bands in Figures 5,7 represent 95% confidence intervals (CI) indicating a range of plausible values for the population mean.

Keyboard shortcut use per menu: Figure 4 shows that the confidence intervals of the 4 menus are large and with a high degree of overlap. Thus we can not draw conclusions on whether a menu layout out-performs the others in promoting *correct keyboard shortcuts*. Figure 5 shows the learning speed per menu plotting the *correct keyboard shortcuts* rate and command selection time per block. Results indicate that for the first 20 blocks BASELINE under-performs to the other menus in terms of *correct keyboard shortcuts* rate. After block 20, differences among the 4 menus appear negligible.

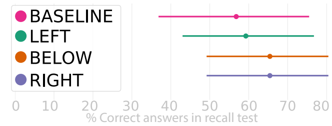

Figure 8: Mean% of recalls per menu.

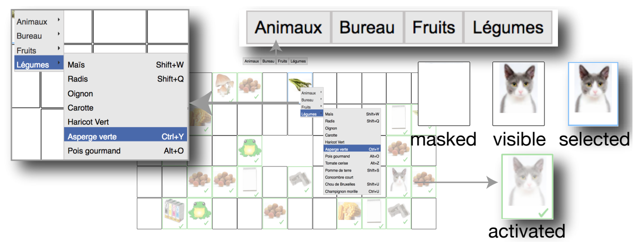

Figure 9: Experiment 2: Interface with 4 target stages

Keyboard shortcut use per command length: BASELINE and RIGHT are sensitive to length of the command, because its length affects the distance between command and shortcut. Thus, we analysed the keyboard shortcut adoption for each command length. Figure 6 shows no conclusive effect for the menu-command length pairs [20]. Yet, we notice a small trend favoring the SMALL and LONG for the BASELINE and RIGHT. Similarly, in Figure 7, shows results of learning speed for each command length which remained inconclusive for both *correct keyboard shortcuts* rate and selection time.

Recall test: For the keyboard shortcut recall test, Figure 8 shows no conclusive differences among the 4 menus.

5 EXPERIMENT 2

Experiment 1 was inconclusive on the difference among the 4 menus. We considered several explanations including that (I) the effect is smaller than expected, or that (II) both the instructions and the task were too explicit on prompting participants to learn shortcuts, masking potential implicit learning behavior, or that (III) the within-subjects design introduced unintended skill transfer, one menu pattern motivating participants to adopt the shortcuts with the other. We run a second experiment to mitigate those possibilities.

To further increase statistical power (I),we compared only two conditions, BASELINE and LEFT, the best rated by designers in terms of noticeability of the shortcut (Figure 3) and followed between-subjects design (III). To improve ecological validity within the ISL context (II), Experiment 2 builds on Banovic et al. design [12] with instructions which did not explicitly invite participants to learn keyboard shortcuts. While this scenario was more realistic, the risk was that a large proportion of participants might not use keyboard shortcuts (lack of awareness, motivation, etc.). We mitigated this risk with the use of safeguards in periodic times during the experiment, using usage tips. Experiment 2 is identical to Experiment 1 including participant payment, apparatus keyboard shortcuts, identical menu hierarchy structure and target items. We next detail only the elements that differ between the two experiments.

5.1 Experimental design

Participants: We recruited 72 different university students (20-30 years) from various fields (e.g. engineering, law, medicine). The duration was 1 hour.

Stimulus & Task: Due to the between-subjects design (III), *all* submenus now had the same layout (all BASELINE or all LEFT). To increase similarity with real applications (II), we added a context menu as a common method to select commands. The new interface (Figure 9) displayed a 5x12 grid in the center of the screen. Each cell had four states: *masked*, where no image is shown; *visible*, the image indicates the target command, i.e. the command that the user has to execute on this cell; *selected*, the image is outlined in blue when clicking on it (with the left or right mouse button). Then they can use their preferred method to select the command (menubar, context menu, or shortcuts); *activated*, the image is blurred, checked and outlined in green to indicate a successfully selected command. The targets appeared in a cell inside the grid interface (Figure 9). Participants had to select a cell before executing the command and then select the next cell, until all cells were activated. The rational behind this design choice was to use a high level task (*"complete the grid"*), rather than series of low-level tasks (execute individual commands). Forcing participants to interact with multiple objects-of-interest (cells) located at different places on the screen can potentially foster more realistic mouse behavior (II). Moreover, it may help participants to perceive the *"real cost"* [21] of mouse-based commands. Indeed, the *"real cost"* of menu-based methods [21] includes not only the time to reach the menu widget (e.g. menubar) and the time to navigate in the menu system, but also the time to *return back* to the objects-of-interest [21] (refer to [7] for extended discussion).

Frequency: Unlike Experiment 1 (uniform distribution), to reflect real application usage (II), we used a zipfian distribution for the frequency of appearance of each target [60]. We used the standard zipfian distribution equation [42, 62] (exponent=1) and applied it to the set of the 12 target commands of Experiment 1 following the procedure described in [32]. The resulted frequencies were rounded off and consisted of: 12, 12, 6, 6, 4, 4, 3, 3, 2, 2, 2, 2. For each session, each item was randomly assigned a frequency.

Procedure: In the pre-test, participants executed again a few simple shortcuts, but this time the keyboard was not connected to the screen. Participants were then told to complete the grid by executing the corresponding commands on each image, but they were not informed about the available methods. So, they did not receive incentives favoring keyboard shortcuts. Between blocks a dialog encouraged them to take a 20 second break. To avoid having too many participants that do not transition we introduced *tips*. From pilots and Experiment 1 data analysis, we saw that participants are very likely to *"never"* transition, if they do not transition during the first three blocks. Therefore, after the third block, we added tips during the inter block breaks (like the ones we can find in modern applications, e.g. Pycharm [2]) to encourage the use of keyboard shortcuts. We added 3 tips appearing at different times. The 1st tip (inter-block 5) informed participants that there are three methods to execute commands (menubar, context menu and keyboard shortcuts). The 2nd tip (inter-block 6) informed them that studies have shown that keyboard shortcuts are faster than menus. The 3rd tip (inter-block 7) explicitly informed them that they should use keyboard shortcuts to optimize their performance. Once a tip was shown, it remained visible during all the following inter-blocks. Such tips ensured having enough data to analyse, while allowed us to capture an initially more spontaneous behaviour. At the end

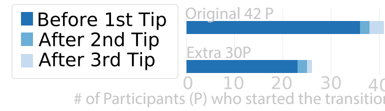

Figure 10: Transitioning to shortcuts after each tip

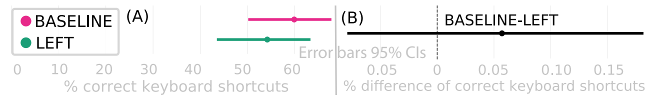

Figure 11: A:Mean% of shortcuts per menu B: Their mean difference

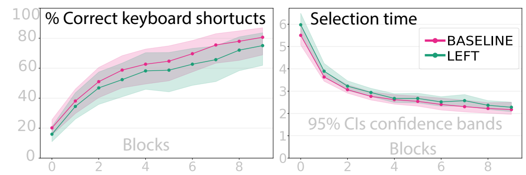

Figure 12: Mean% of shortcuts & time per menu

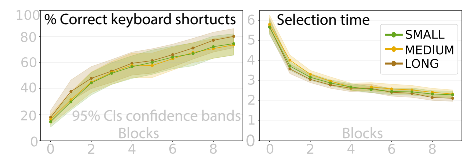

Figure 13: Mean% of shortcuts & time per command length

of the session, participants filled a questionnaire and preformed a recall test similar to Experiment 1.

Design: We used a mixed-subjects design, with participants randomly assigned to one *menu* (BASELINE or LEFT). The within design factor was the command frequency. The position of targets in the grid was randomly assigned in each block. Each participant performed 10 blocks of 58 selections. Overall, the design was 2 menus × 36 participants × 10 blocks × 58 trials = 41760 selections.

Planned analysis: Our original planned analysis was to recruit 42 participants who met the criteria threshold to user *correct keyboard shortcuts* for at least 25% of the trials. We based this decision on [41] who used a similar protocol to ensure that they had enough participants used shortcuts. However, we finally decided to fully report the same analysis of Experiment 1. We also realised that 42 participants is a rather arbitrary number which might not be enough to capture the effect [5]. Therefore we added 30 participants based on the formula of Prashant et al. [36]. This formula calculates the sample size for a follow-up experiment based on the results of the initial experiment. We considered those two deviations acceptable, given that the new analysis includes the planned analysis.

5.2 Results

Errors bars in Figures 11, 15, 14, 16 and confidence bands in Figures 12, 13 represent 95% confidence intervals (CI).

Tips: Figure 10 shows when participants started the transition (i.e., used *correct keyboard shortcuts* for the first time) based on the different tips. The bars show the results of the 42 participants and the additional 30 participants. We observe that the majority of the participants started using *correct keyboard shortcuts* before the

[5] In the osf.io link, we provide results for the original planned analysis as well for the 30 added participants separately in which the effects were consistent with the current reported analysis.

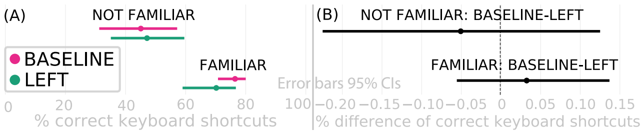

Figure 14: A:Mean% of shortcuts per menu and familiarity B: Their mean difference.

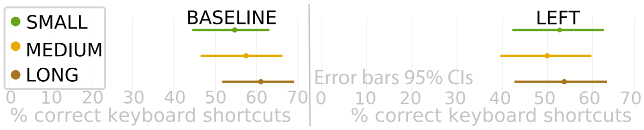

Figure 15: Mean% of shortcuts per command length

Figure 16: Mean% of recalls per menu for all (A), FAMILIAR (B) and UNFAMILIAR (C) participants.

1st tip (82%), indicating the lack of influence of the tips on the awareness of the keyboard shortcuts for most participants.

Keyboard shortcut use menu: Figure 11 reports the results of our analysis regarding the differences between the two layouts.Our analysis indicates that the mean rate difference is inconclusive between the 2 layouts. Figure 12 reports the results of the learning speed analysis. We observe a trend favoring the BASELINE over the LEFT especially regarding the mean rate adoption per block.

Keyboard shortcut use per menu item length: Similar to Experiment 1 we investigated the effect of the command length for each layout on the keyboard shortcut adoption. Figure 15 shows no conclusive difference among the menu-command length pairs. Figure 13 reports the results of the learning speed analysis. Once again, it remains unclear if the learning speed was affected by the three conditions.

Keyboard shortcut use per user profile: To better understand our results we decided to perform an extra analysis based on the users' profiles. In particular we are interested to see if their familiarity with keyboard shortcuts played a role in their behavior. In the post-questionnaire of each session in both studies we asked the participants to indicate their familiarity with keyboard shortcuts. Most Experiment 1 participants (21 out of 26) were UNFAMILIAR with keyboard shortcuts. In contrast, in Experiment 2, the ratio between the FAMILIAR-UNFAMILIAR users was better (44 FAMILIAR, 28 UNFAMILIAR). We thus decided to investigate where their profiles may have affected our results. Figure 14 reports the results of the keyboard shortcut adoption per layout(BASELINE, LEFT) and user profile(FAMILIAR, UNFAMILIAR). Results indicate that FAMILIAR users used more keyboard shortcuts than the UNFAMILIAR but we couldn't detect any differences among layouts and user profiles

Recall test: Figure 16 shows the mean rate of correct answers for each menu layout for all participants and for each familiarity group. We couldn't detect any differences among the 2 layouts.

6 DISCUSSION & FUTURE DIRECTIONS

Although our results did not confirm our original hypothesis, we argue that they are informative because, based on our review of ISL theory they are surprising and open a number of future directions:

Familiarity & User Profile. While the familiarity analysis (Figure 14 and 16) did not produce conclusive results, we observed an interesting trend. The FAMILIAR group performed better with BASELINE than LEFT while participants in the UNFAMILIAR group performed better with LEFT than BASELINE. It is worth investigation whether interactions between familiarity and spatial relationships influence implicit learning. It is likely that the FAMILIAR group exploits prior experience with the menu pattern and how its elements are positioned, while the UNFAMILIAR group lacks this advantage.

Other ISL Factors. In the ISL related work, we identified further design factors worth investigating. For instance, **exposure time** may increase the adoption of the keyboard shortcut. In drop-down menus, the commands are only visible when the menu is open which may hinder the repeated exposure of the pattern. Other command selection widgets like command palettes or toolbars could be more appropriate as they are always visible. Another factor is **accumulation** of instances in memory, for which we know that only limited contingencies can be learnt in a restricted period, suggesting that only a few associations trigger learning [55]. In our context, this suggests that limiting the total number of commands can facilitate the implicit learning of the remaining ones.

Methods. It remains unclear how to effectively transpose findings and methodologies from cognitive science to the HCI context. Alternative experimental protocols can be investigated. For instance, 1-hour experiments may not be enough to capture such phenomenona. Some experimental protocols in cognitive science use multiple sessions as exposure time [47, 52, 54]. Future investigations could increase exposure time while an eye-tracker can detect when and how long participants gaze the menu commands.

Traditional ISL studies "force" participants to learn a single pattern [56]. In contrast, our study was based on prior work [32] and let the participant decide which modality they wish to use to complete their task. This decision could have affected the ISL processes. Further investigation could examine the role of choice between different modalities in ISL study design.

Questioning UI design conventions. This paper questioned menu design conventions used in existing systems to some extend. We extracted the design space of spatial relationships in the menu patterns and as future work, we suggest extending this design space considering additional criteria. For instance, through the lens of ISL theory, we can revisit icon and symbol (e.g. ►) placements and radically different menu patterns (e.g. circular layout [3]). It is also worth investigating the influence of spatial relationships on other aspects of usability (readability, aesthetics, preferences etc). Our study with professional designers (Figure 3) suggests potential benefits of alternative menu layouts (e.g BELOW) on usability (e.g. aesthetics). A common barrier for practitioners in questioning conventions is to confuse users hindering, at least temporally, their performance. Our findings showed that it is possible that *less conventional menu designs do not impact negatively user performance even within limited exposure.* We thus encourage HCI research to be open in revisiting such conventions and investigate how design novelty can benefit users.

REFERENCES

[1]]hickeys [n. d.]. https://shorturl.at/pqKNR
[2] 2018. Pycharm: Python IDE for Professional Developers.
[3] David Ahlstroem, Rainer Alexandrowicz, and Martin Hitz. 2006. Improving menu interaction: a comparison of standard, force enhanced and jumping menus. *ACM CHI* 1 (2006), 1067–1076.

[4] Caroline Appert and Shumin Zhai. 2009. Using Strokes As Command Shortcuts: Cognitive Benefits and Toolkit Support. In *ACM CHI*. 2289–2298.
[5] Apple Cooporation. 2018. . https://developer.apple.com/macos/human-interface-guidelines/windows-and-views/toolbars/
[6] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2009. MenUA: a design space of menu techniques. *Retrieved April* 2 (2009), 2016.
[7] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2016. Visual Menu Techniques. *Comput. Surveys* 49, 4 (2016), 1–41.
[8] Gilles Bailly and Sylvain Malacria. 2013. MenuInspector: Outil pour l'analyse des menus et cas d'étude. In *ACM IHM*. 103.
[9] Gilles Bailly, Khamassi Mehdi, and Benoit Girard. 2022. Computational Model of the Transition from Novice to Expert Interaction Techniques. *ACM ToCHI. to appear* (2022).
[10] Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. 2013. MenuOptimizer: interactive optimization of menu systems. In *ACM UIST*. 331 – 342.
[11] Chris Baker, Carl Olson, and M. Behrmann. 2004. Role of attention and perceptual grouping in visual statistical learning. *Psychol.Sci.* 15, 7 (2004), 460–466.
[12] Nikola Banovic, F. Chevalier, Tovi Grossman, and G. Fitzmaurice. 2012. Triggering triggers and burying barriers to customizing software. In *ACM CHI*. 2717–26.
[13] Ana Caraban, Evangelos Karapanos, Daniel Gonçalves, and Pedro Campos. 2019. 23 ways to nudge: A review of technology-mediated nudging in human-computer interaction. In *ACM CHI*. 1–15.
[14] James Carpenter and John Bithell. 2000. Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians. *Statistics in medicine* 19, 9 (2000), 1141–1164.
[15] Tony F Chan, Gene H Golub, and Randall J LeVeque. 1983. Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician* 37, 3 (1983), 242–247.
[16] Morten H Christiansen. 2019. Implicit statistical learning: A tale of two literatures. *Topics in cognitive science* 11, 3 (2019), 468–481.
[17] Marvin M Chun and Yuhong Jiang. 1998. Contextual cueing: Implicit learning and memory of visual context guides spatial attention. *Cog.Psy* 36, 1 (1998).
[18] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2014. Supporting Novice to Expert Transitions in User Interfaces. 47, 2 (2014), 1–36.
[19] Christopher M Conway and Morten H Christiansen. 2006. Statistical learning within and between modalities: Pitting abstract against stimulus-specific representations. *Psychol.Sci.* 17, 10 (2006), 905–912.
[20] Geoff Cumming and Sue Finch. 2005. Inference by eye: confidence intervals and how to read pictures of data. *American Psychologist* 60, 2 (2005), 170.
[21] R. F. Dillon, Jeff D. Edey, and Jo W. Tombaugh. 1990. Measuring the True Cost of Command Selection: Techniques and Results. In *ACM CHI*. 19–26.
[22] Pierre Dragicevic, Fanny Chevalier, and Stéphane Huot. 2014. Running an hci experiment in multiple parallel universes. In *ACM CHI Ext Abstracts*. 607–618.
[23] Nick C Ellis. 2015. Implicit and explicit learning: Their dynamic interface and complexity. John Benjamins.
[24] Nobutaka Endo and Yuji Takeda. 2005. Use of spatial context is restricted by relative position in implicit learning. *Psychon Bull Rev* 12, 5 (2005), 880–885.
[25] Leah Findlater, Karyn Moffatt, Joanna McGrenere, and Jessica Dawson. 2009. Ephemeral adaptation: The use of gradual onset to improve menu selection performance. In *ACM CHI*. 1655–1664.
[26] Andrea L Gebhart, Elissa L Newport, and Richard N Aslin. 2009. Statistical learning of adjacent and nonadjacent dependencies among nonlinguistic sounds. *Psychon Bull Rev* 16, 3 (2009), 486–490.
[27] Emmanouil Giannisakis, Gilles Bailly, Sylvain Malacria, and Fanny Chevalier. 2017. IconHK: Using Toolbar button Icons to Communicate Keyboard Shortcuts. *ACM CHI* (2017), 4715–4726.
[28] Fernand Gobet, Peter CR Lane, Steve Croker, Peter CH Cheng, Gary Jones, Iain Oliver, and Julian M Pine. 2001. Chunking mechanisms in human learning. *Trends Cogn.Sci.* 5, 6 (2001).
[29] Alison Gopnik and Henry M Wellman. 2012. Reconstructing constructivism: Causal models, Bayesian learning mechanisms, and the theory theory. *Psychol. Bull.* 138, 6 (2012), 1085.
[30] A. Goujon, A. Didierjean, and S. Poulet. 2014. The emergence of explicit knowledge from implicit learning. *Mem Cognit* 42, 2 (2014), 225–236.
[31] A. Goujon, A. Didierjean, and S. Thorpe. 2015. Investigating implicit statistical learning mechanisms through contextual cueing. *Trends Cogn.Sci.* 19, 9 (2015).
[32] Tovi Grossman, Pierre Dragicevic, and Ravin Balakrishnan. 2007. Strategies for Accelerating On-line Learning of Hotkeys. In *ACM CHI*. 1591–1600.
[33] Tovi Grossman, George Fitzmaurice, and Ramtin Attar. 2009. A survey of software learnability: metrics, methodologies and guidelines. In *CHI*. 649–658.
[34] Marshall M Haith, Naomi Wentworth, and Richard L Canfield. 1993. The formation of expectations in early infancy. *Advances in infancy research* (1993).
[35] Mary Hegarty, Patricia A Carpenter, and Marcel Adam Just. 1991. Diagrams in the comprehension of scientific texts. (1991).
[36] Prashant Kadam and Supriya Bhalerao. 2010. Sample size calculation. *International journal of Ayurveda research* 1, 1 (2010), 55.
[37] Elmar Kal, Rens Prosé, Marinus Winters, and John van der Kamp. 2018. Does implicit motor learning lead to greater automatization of motor skills compared to explicit motor learning? A systematic review. *PloS one* (2018).
[38] Elma Kerz, Daniel Wiechmann, and Florian B Riedel. 2017. Implicit learning in the crowd: Investigating the role of awareness in the acquisition of L2 knowledge. *Studies in Second Language Acquisition* 39, 4 (2017), 711–734.
[39] Brian Krisler and Richard Alterman. 2008. Training Towards Mastery : Overcoming the Active User Paradox. (2008), 18–22.
[40] Gordon Paul Kurtenbach. 1993. *The design and evaluation of marking menus*. Ph. D. Dissertation. University of Toronto.
[41] B Lafreniere, C Gutwin, and A Cockburn. 2017. Investigating the post-training persistence of expert interaction techniques. *ACM TOCHI* 24, 4 (2017), 29.
[42] Wanyu Liu, Gilles Bailly, and Andrew Howes. 2017. Effects of Frequency Distribution on Linear Menu Performance. In *ACM CHI*. 1307–1312.
[43] Robert L Mack and Jakob Nielsen. 1995. Usability inspection methods: Executive summary. In *Readings in Human–Computer Interaction*. Elsevier, 170–181.
[44] S Malacria, G Bailly, J Harrison, A Cockburn, and C Gutwin. 2013. Promoting Hotkey Use Through Rehearsal with ExposeHK. In *ACM CHI*. 573–582.
[45] Sylvain Malacria, Joey Scarr, Andy Cockburn, Carl Gutwin, and Tovi Grossman. 2013. Skillometers: Reflective Widgets That Motivate and Help Users to Improve Performance. In *ACM UIST*. 321–330.
[46] JP Maxwell, RSW Masters, E Kerr, and E Weedon. 2001. The implicit benefit of learning without errors. *Q J Exp Psychol* 54, 4 (2001), 1049–1068.
[47] Pierre Perruchet and Sebastien Pacton. 2006. Implicit learning and statistical learning: One phenomenon, two approaches. *Trends Cogn.Sci.* 10, 5 (2006).
[48] Pierre Perruchet and Annie Vinter. 1998. PARSER: A model for word segmentation. *Journal of memory and language* 39, 2 (1998), 246–263.
[49] Arthur S Reber. 1989. Implicit learning and tacit knowledge. *J.Exp.Psychol.Gen.* 118, 3 (1989), 219.
[50] Jenny R Saffran, Elissa L Newport, Richard N Aslin, Rachel A Tunick, and Sandra Barrueco. 1997. Incidental language learning: Listening (and learning) out of the corner of your ear. *Psychol.Sci.* 8, 2 (1997), 101–105.
[51] Jenny R Saffran and Erik D Thiessen. 2007. Domain-general learning capacities. *Blackwell handbook of language development* (2007), 68–86.
[52] Joey Scarr, Andy Cockburn, Carl Gutwin, and Philip Quinn. 2011. Dips and Ceilings: Understanding and Supporting Transitions to Expertise in User Interfaces. In *ACM CHI*. 2741–2750.
[53] Ryan B Scott and Zoltán Dienes. 2008. The conscious, the unconscious, and familiarity. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 34, 5 (2008), 1264.
[54] Dennis Shelton and Robert C Newhouse. 1981. Incidental learning in a paired-associate task. *Int. J. Exp. Educ* 50, 1 (1981), 36–38.
[55] Andrea C Smyth and David R Shanks. 2008. Awareness in contextual cuing with extended and concurrent explicit tests. *Mem Cognit* 36, 2 (2008), 403–415.
[56] Cyril Thomas, A. Didierjean, F. Maquestiaux, and Annabelle Goujon. 2018. On the limits of statistical learning: Intertrial contextual cueing is confined to temporally close contingencies. *Atten Percept Psychophys* (2018), 1–16.
[57] Juan M Toro, Scott Sinnett, and Salvador Soto-Faraco. 2005. Speech segmentation by statistical learning depends on attention. *Cognition* 97, 2 (2005), B25–B34.
[58] Theophanis Tsandilas and MC Schraefel. 2007. Bubbling menus: a selective mechanism for accessing hierarchical drop-down menus. In *ACM CHI*. 1195–1204.
[59] Nicholas B Turk-Browne, Justin A Jungé, and Brian J Scholl. 2005. The automaticity of visual statistical learning. *J.Exp.Psychol. Gen.* 134, 4 (2005), 552.
[60] Ian H. Witten, John G. Cleary, and Saul Greenberg. 1984. On frequency-based menu-splitting algorithms. *IJMMS* 21, 2 (1984), 135–148.
[61] Xuelian Zang, Lina Jia, Hermann J Müller, and Zhuanghua Shi. 2015. Invariant spatial context is learned but not retrieved in gaze-contingent tunnel-view search. *JoEP: Learning, Memory, and Cognition* 41, 3 (2015), 807.
[62] George Kingsley Zipf. 2016. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books.