Cloud Menus, a Circular Adaptive Menu for Small Screens

Jean Vanderdonckt[1], **Sara Bouzit**[2,3], **Gaëlle Calvary**[3], **Denis Chêne**[2]
[1]Université catholique de Louvain, [2]Orange Labs, [3]Université Grenoble Alpes
jean.vanderdonckt@uclouvain.be, {sara.bouzit, gaelle calvary}@imag.fr, denis.chene@orange.com

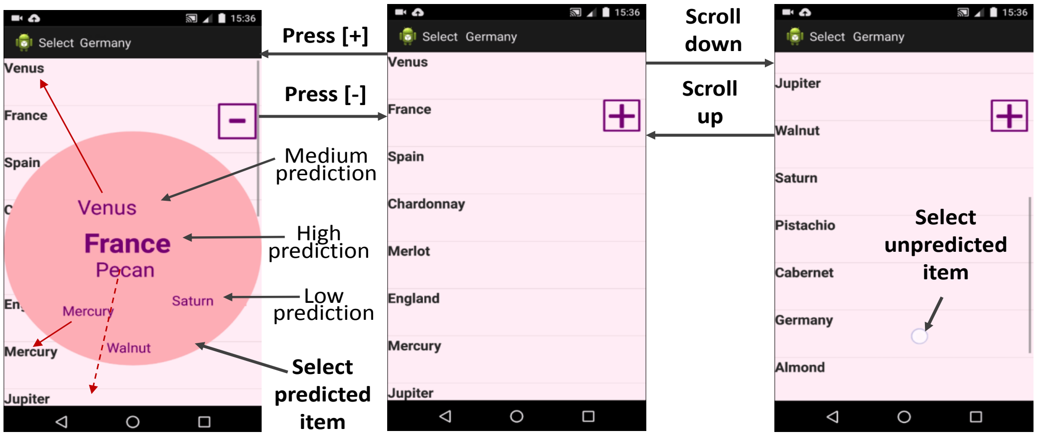

Figure 1. The Cloud Menus: a circular adaptive cloud contains predicted items that can be made (in)visible by pressing [+], [–].

ABSTRACT

This paper presents Cloud Menus, a split adaptive menu for small screens where the predicted menu items are arranged in a circular tag cloud with a location consistent with their corresponding position in the static menu and a font size depending on their prediction level. This layout results from a 3-step design process: (i) defining an initial design space on Bertin's 8 visual variables and 4 quality properties, (ii) identifying the most preferred layout based on agreement rate, and (iii) implementing it into Cloud Menus, a new widget for Android with circular layout. An empirical study suggests that cloud menus reduce item selection time and error rate when prediction is correct without penalizing it when prediction is incorrect, compared to two baselines: a non-adaptive static menu and an adaptive linear menu. From this study, design guidelines for cloud menus are elaborated.

Author Keywords

Adaptive menu, prediction window, split menu, tag cloud.

ACM CLASSIFICATION KEYWORDS

- **Human-centered computing ~ Human computer interaction (HCI)**.

INTRODUCTION

Split Adaptive Interfaces [11] reproduce a limited part of the User Interface (UI) elements (e.g., a menu [30], a toolbar [15], icons [19,26], a help system [1]) that are predicted to be of immediate use to preserve stability [13]. Split adaptive menus typically present the end user with predicted menu items for improving feature findability [13], speeding up the item selection [9] without searching for them in deep and wide menus. Since selecting a menu item requires a navigation time and a visual search time that depend on the number of items displayed, small screens are particularly affected and the risk of not benefiting from the adaptivity advantages is real. UI adaptation remains essential to ensure the user experience and to introduce the right UI changes at the right time [20]. The quality of adaptation is proportional to the quality of prediction [14,32]: if the prediction is accurate enough, adaptation brings its earnings by accelerating and facilitating user interaction. Since the quality of prediction can never be guaranteed, the following questions are stated: what do we win when the prediction is accurate?, how to avoid penalizing interaction when prediction is wrong?, how to prevent user from errors induced by a wrong prediction? In order to address these challenges, this paper proposes *Cloud Menus* (Fig. 1), a split adaptive menu for small screens with a prediction window displayed as a word cloud and demonstrates that it accelerates interaction when prediction is correct without slowing it down when prediction is incorrect. Also important is to assess to what extent increasing the amount of predicted items impacts the interaction.

The remainder of this paper is structured as follows: Section 2 reviews works related to adaptive menus for small screens. Section 3 discusses the 3-step design process that leads to defining and implementing Cloud Menus. Section 4 reports on a controlled study performed to test Cloud Menus (with adaptivity) against a static menu (without any adaptivity) and a linear menu (with adaptivity,

IUI 2018, March 7–11, 2018, Tokyo, Japan.
© 2018 ACM ISBN 978-1-4503-4945-1/18/03...$15.00.
https://doi.org/10.1145/3172944.3172975

but without the cloud layout). Section 5 concludes the paper, by presenting some future avenues to this paper.

RELATED WORK

There is a rich background [2,3] of designing interaction techniques for menu selection, which is understood as one of the most frequently used interaction style in UIs. While a great deal of this work focuses on the interaction techniques themselves, prior work also investigates how a menu could be made adaptive depending on different data sources.

Adaptive Prompting [19] predicts a selection of applications and related files based on an application model (which specifies relationships between elements) and a user model (which manually specifies the user's experience level, etc.) to reduce navigation effort. How applications and files are presented is also subject to adaptivity.

Frequency-based menus sort menu items by decreasing order of frequency, depending on the end user's actions [2]. *Split menus* [30] separate a graphical menu into a topmost area promoting a small list of 2-3 predicted items and a static menu containing non-predicted items. Split menus are subject to instability [9,13]: end users oscillate between the two areas because they are unsure where to find the menu item they want. To preserve this stability, *Split menus with replication* leaves the adaptive part as stated before and the second part remains unaltered, thus enabling end users to always refer to the menu as they know it. *Adaptive Activation-Area Menu* (AAMU) [33] is an adaptive morphing menu containing an enlarged activation area for predicted items which dynamically resizes itself providing a broader steering path for menu navigation. Predicted items can be emphasized by highlighting [1], bolding [24], coloring or underlying [21], changing font size [33], moving them [23] to a comfort zone [25], by shortcut [7], and animation [10,26].

Ephemeral menus [10] is an adaptive temporal menu where the gradual onset was used in order to display non-predicted items. At opening the menu, user finds predicted items and after a delay of 500ms remaining items appear gradually. When the prediction accuracy is relatively high (79%), ephemeral adaptation significantly offers better performance and user satisfaction over static menus, and a performance benefit over a graphical adaptation technique. Obtaining high prediction accuracy of menu items is crucial to the success of adaptive menus. However, there still lacks an effective prediction method of menu items. *In-Context Disappearing* (ICD) are smartphone menus [5]: at opening the initial menu, the full list of items is superimposed with a window prompting predicted items. This latter contains three items and disappears gradually. *Out-of-Context Disappearing (OCD)* approach is the inverse: at opening, the prediction window is immediately displayed with the predicted items; after a delay of 500 msec [10], the complete menu is gradually displayed from the back, thus replacing the prediction window.

The perceived efficiency of three split menus is measured [9]: a static menu with top four items remaining constant, an adaptable menu where top four items can be changed by the end user, and an adaptive menu where top four items in the split menu are predicted according to user's recently and frequently used items. Static menus are found efficient compared to both adaptable and adaptive menus. However, overall participants tend to prefer adaptable menus.

Split Adaptive Interfaces exhibit a certain amount of potential benefits at a certain cost [14,20]. The gain expected from promoting predicted items should be counter-balanced by the end user's need to constantly adapt to the altered menu layout, especially for novice users who hesitate between areas. The gain becomes positive when adaptivity significantly reduces the menu selection time, such as in a hierarchical menu [13,34] or when limited screen resolution induces scrolling [32]. Other potential shortcomings are: adaptive menus do not work well with short menus [27] or when the end user alternates between an amount of items that is larger than those contained in the prediction window [13].

Bridle & McCreath [6] predict menu selection on a mobile phone by relying on machine learning to improve usability: it helps to reduce the number of key presses. While Xie *et al.* [35] predict menu items on a mobile phone based on a Markov Chain, Fukazawa *et al.* [12] prefer Support Vector Machine (SVM): the system ranks both frequently and rarely used functions based on user operation history. While accessibility of these menu items is improved, this system may cause significant performance loss [12]. Asthana *et al.* [2] study the adverse impact of adaptive voice menu on experienced users. They also propose strategies to lower the negative effect of adaptivity on familiar users. Gajos & Chauncey [15] demonstrate systematic individual differences in the utilization of the adaptivity, which correlate with the stable user traits of Need for Cognition and Extraversion.

Stability can be further refined based on Bertin's height *visual variables* [4] conveying an adaptation scheme [5]: *position* (e.g., change in the *x*, *y*, *z* location), *size* (e.g., change in length, area, or repetition), *shape* (e.g., change by shape), *value* (e.g., change from light to dark), *orientation* (e.g., change in alignment, angle), *color* (e.g., change in hue at a given value), *texture* (e.g., change in pattern), and *motion* (e.g., animated transition or visual effect). Four properties can be defined for comparing adaptive menu [5]:

1. *Spatial stability*: the ability of an adaptive menu to preserve its spatial layout after adaptivity, thus keeping position and orientation constant.
2. *Physical stability*: the ability of an adaptive menu to preserve its physical configuration after adaptivity, thus keeping size and shape constant.
3. *Format stability*: the ability of an adaptive menu to preserve the format of its layout after adaptivity, thus keeping value, color, and texture constant.

4. *Temporal stability*: the ability of an adaptive menu to preserve its layout over time while being adapted, thus keeping motion constant.

For example, ephemeral adaptation [10] preserves spatial, physical, and format stability, but not temporal stability.

In conclusion, several adaptive menus have been researched for pull-down menus displayed on desktop [34]. Transposing these techniques to small screens is not straightforward because all items cannot be displayed on a single screen [27] and because they require interaction techniques that small screens are not able to afford [18]. These adaptive menus are always presented at once, which is not the case on small screens: any predicted item located on a subsequent screen requires a cognitive effort to explore the whole set of items.

There are not many adaptive menus tailored to small screens. Either menus are adaptive, but not tailored to small screens or menus are tailored for these devices, but pursue different goals, such as information display optimization [21], reducing number of taps [27], increasing efficiency [27]. The amount of predicted items in the aforementioned menus is usually limited to 2-3 items [10,14]. Increasing this amount has never been explored before and the presentation of the prediction window remains very similar to the initial menu.

THE CLOUD MENU DESIGN PROCESS

An exploratory session was organized by an Internet Service Provider with ten participants, two UI designers, one Human Factors expert and one experienced developer. From a focus group, five requirements were elicited for a new adaptive menu for smartphones: (**R1**) provide an adaptive menu that accelerates user interaction when item prediction is accurate without decelerating it when prediction is inaccurate, (**R2**) address constraints imposed by small screen devices, (**R3**) display more than 3 predicted items if possible (as opposed to other adaptive menus), (**R4**) maintain spatial stability, and (**R5**) rely on physical, format, and temporal coding. Then, a 3-step design process has been conducted in order to progressively define and investigate a design space.

Step 1: Define a Design Space of Prediction Window

The goal of this design space is to define how the prediction window could be laid out by theoretically exploring design alternatives. Bertin's visual variables offer eight dimensions for choosing the layout of the presentation window, but have a varying ability to be *selective* (is a change enough to allow us to select it from a group) and/or *ordered* (are changes according to this variable perceived as ordered?) (Table 1). Fig. 2 sorts coding schemes by decreasing order of precision: in the last column for nominal values (we hereby assume that menu items are neither quantitatively sorted nor ordered), position is the most precise coding scheme, length and surface coming later, the other ones have not been retained.

	Bertin's visual variable [3]	Select	Order
1	Position	!	!
2	Size	!	!
3	Shape	3	1
4	Value	!	!
5	Color	3	1
6	Orientation	!	1
7	Texture	!	1
8	Motion	3	1

Table 1. Level of support of Bertin's visual variables expressed according to Harvey's Balls.

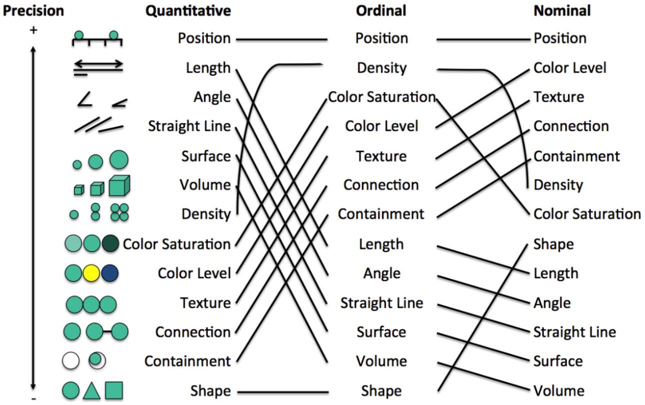

Figure *2*. Coding schemes sorted by order of precision.

The Cloud Layout for Prediction Window

Addressing R5, while respecting R4, means that the prediction window should not necessarily be embedded in the initial menu, but could be presented in another way, such as by superimposition, but not far away. Therefore, the driving principle of the prediction window layout consists in displaying a *modal word-cloud based* [17] *prediction window* overlapping the full initial menu and enabling the user to control the prediction window on-demand. The general cloud layout was initially motivated by the following reasons.

A *tag cloud*, or *Wordle* [8], is a visual representation of text data, typically used to depict keyword metadata (*tags*) on any textual document or collection of documents (e.g., a report, a web site, a set of papers, a digital library) [22]. Tags are usually single words whose importance is shown is depicted through one or many visual variables [4], typically font size or color. These coding schemes are useful for quickly perceiving the most prominent terms and for locating them to determine their relative prominence. Tag clouds are extensively researched and used in many areas [8,16,17,22,28,31], except for menu selection: for *indexing* (e.g., for faster term discriminativeness), for *searching* (e.g., semantic expansions for web search, personalized search), for *generating a taxonomy* (e.g., for generating a taxonomy of terms selected in a domain), for *clustering and classification* (e.g., classifying blog entries and general web objects), for *social interest discovery* (e.g., user profiling, end user's topics of interest), and *browsing* (e.g., when terms are hyperlinked to items associated with the tag). *Word clouds* [17] emerge as a usable [28], straightforward and visually appealing [31] method for emphasizing important terms in a textual document or a collection of documents. They are used in various contexts as a means to provide an overview by drilling down text to those terms that appear with the highest frequency. The word cloud layout has an impact on its perceived usability [22,31].

Halvey & Keane [16] examine the time for selecting a country a list of all countries presented in 6 varying formats: alphabetical horizontal list (μ=2.88 sec), alphabetical vertical list (μ=2.89 sec), alphabetical cloud (μ=2.94 sec), horizontal list (μ=3.19 sec), vertical list (μ=3.24 sec), and normal cloud (μ=3.40 sec). The cloud is slower than both lists because items in the lists are ordered alphabetically (which is faster with westernized reading) as opposed to distance from the center (which is faster when scanning). When the font size of items in the cloud is larger, the cloud reaches to μ=2.74 sec. These results suggest that a cloud-based representation for a menu, where items are not necessarily sorted by alphabetical order, may be appropriate since it does not decrease the performance with respect to lists. The question of which cloud layout then arises for the prediction window.

Findlater test

In order to explore cloud representations based on Bertin's visual variables, the menu considered by Findlater *et al.* [10] for their experiment on *ephemeral adaptation* was considered: the full menu contains 4 groups of 4 related items (i.e., England, France, Germany, Spain – Venus, Mercury, Jupiter, Saturn – Cabernet, Chardonnay, Merlot, Shiraz – Almond, Pecan, Pistachio, Walnut) and the prediction was defined as follows: Venus=80%, Spain, Shiraz=70%, Pecan, Cabernet, Pistachio=60%, all other items having the same normal probability to be selected. This configuration is in line with other research in this area [5,34] both in terms of prediction probabilities and in terms of prediction level [14]. We will refer to this list as the *Findlater test*.

Initial prototypes for Cloud Menus

Based on Findlater test, initial prototypes for Cloud Menu layouts were developed in Adobe Flash V27.0 for Bertin's variables: *size* (1D vertical vs 1D horizontal vs 2D – Fig. 3), *shape* (rectangle, circle, oval – Fig. 4), *value* (highlighting of current item, zooming in, zooming in with rotation in case of a vertical label – Fig. 5), *color* (Fig. 6a), *font size* (Fig. 6b), *orientation* by changing label angle (horizontal, vertical, angular – Fig. 7), *texture* by changing font family (regular, bold, italic, or combined – Fig. 6b), and *motion* by animation (without vs with animation of non-horizontal items – Fig. 8).

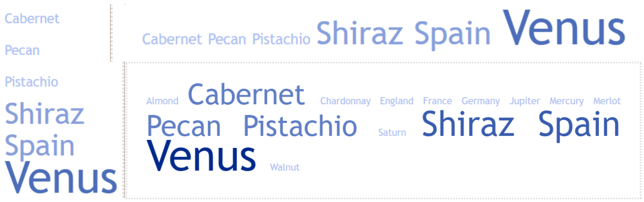
Cabernet
Pecan
Pistachio
Shiraz
Spain
Venus
Cabernet Pecan Pistachio Shiraz Spain Venus
Almond Cabernet Chardonnay England France Germany Jupiter Mercury Merlot
Pecan Pistachio Saturn Shiraz Spain
Venus Walnut

Figure 3. 1D vertical, 1D horizontal, 2D layouts.

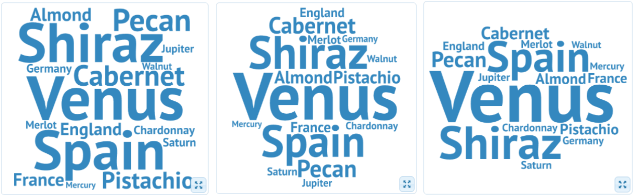
Almond Pecan Shiraz Jupiter Germany Walnut Cabernet Venus Merlot England Chardonnay Saturn Spain France Mercury Pistachio
England Cabernet Merlot Germany Shiraz Walnut Almond Pistachio Venus Mercury France Chardonnay Spain Saturn Pecan Jupiter
Cabernet England Merlot Walnut Pecan Spain Mercury Jupiter Almond France Venus Chardonnay Pistachio Shiraz Germany Saturn

Figure 4. Rectangle, circle, oval layouts.

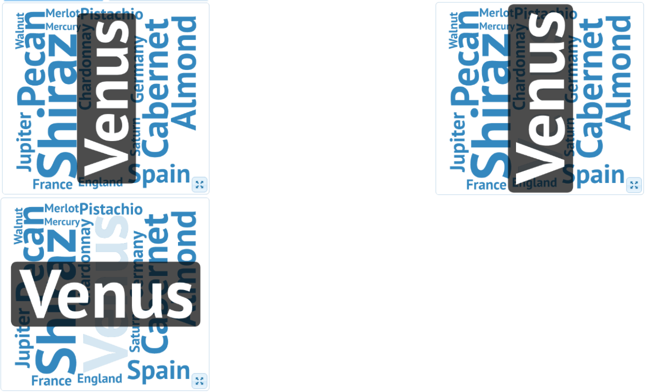
Merlot Pistachio Mercury Walnut Pecan Jupiter Shiraz Chardonnay Venus Germany Saturn Cabernet Almond France England Spain

Figure 5. Layout with highlighting, zooming in, zooming in with rotation of vertical label.

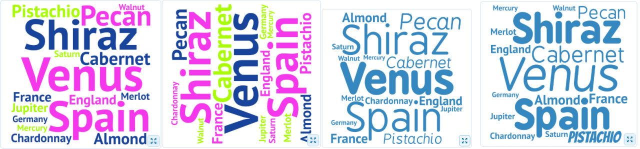
Pistachio Pecan Walnut Shiraz Saturn Cabernet Venus France England Merlot Jupiter Germany Mercury Spain Chardonnay Almond
Almond Pecan Shiraz Saturn Walnut Mercury Cabernet Venus Merlot Chardonnay England Jupiter Germany Spain France Pistachio
Mercury Walnut Pecan Merlot Shiraz England Cabernet Venus Germany Almond France Jupiter Spain Chardonnay Saturn PISTACHIO

Figure 6. Layouts with colors (a), different fonts (b).

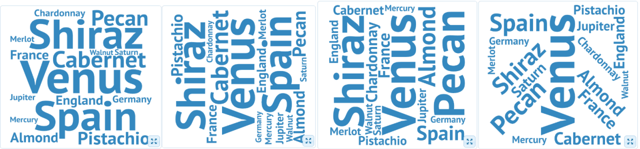
Chardonnay Pecan Merlot Shiraz France Cabernet Venus Jupiter England Germany Mercury Spain Almond Pistachio
Cabernet Mercury Spain Pistachio Jupiter Germany Chardonnay Almond France Venus Shiraz Saturn Pecan Merlot Walnut

Figure 7. Horizontal, vertical, bidirectional, angular layouts.

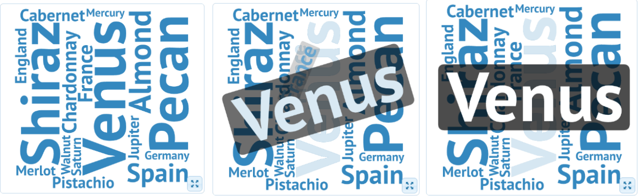
Cabernet Mercury England Shiraz Chardonnay France Venus Walnut Saturn Jupiter Almond Pecan Germany Merlot Pistachio Spain

Figure 8. Layouts with animation of non-horizontal items, without or with zoom of selected item, with adjustable speed.

Step 2: Exploratory Study of Prediction Window Layouts

Similarly to the procedure by Ponsard *et al.* [26], who prototyped various techniques for ephemeral adaptation of icons, we performed an exploratory user study to evaluate people's reactions with respect to the various layouts generated (Fig. 3-8). We were interested in what people would think of the different adaptations conveyed by the different layouts of the prediction window and which they would prefer.

Procedure of the Exploratory Study

Each participant performed the task in a controlled environment. Prior to the task each participant was welcomed, had the process explained to them (including signing a consent form) and filled in a short questionnaire on their background. After the questionnaire was completed, the researcher demonstrated the initial prototypes for Cloud Menus. The participants were given 5 min. to familiarize themselves with the software and ask any questions. If desired, the participant could finish this part early. The participants were then given 15 min. to use a small program where each referent can be tested. The instructions were to select each referent one after another and experiment the corresponding layout. During this experiencing time, the researcher sat next to the participant and observed them. The participant then rated each layout using a five point Likert scale on the layout effectiveness in preserving the meaning: one was strongly disagree through to five being strongly agree. Then they were asked to rank the layouts in order from most preferred to least preferred. After reviewing the referents, the participant was asked for any recommendations on which layout could be selected for conveying the prediction window. This then concluded the user study.

Analysis

After each participant, the questionnaire, ratings and ranking data was added into an MS Excel spreadsheet. The data was entered in an anonymous format so the participants could not be identified.

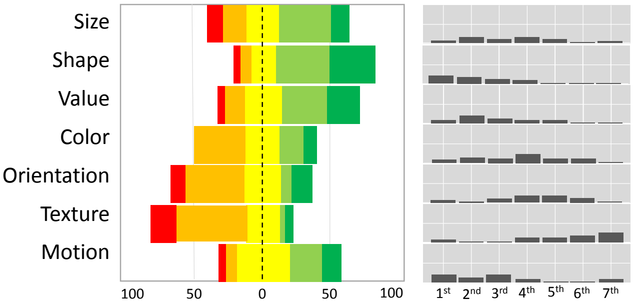

Figure 9. Participants' rating and ranking for each variable.

Results and Discussion

A total of thirty participants (μ=32.3 years, SD=6.2 y.) participated in this experiment. All participants were regular computer users and recruited in our organization through a mailing list. They have different backgrounds such as: accounting, finance, information systems, management, marketing, and human resources. They were all volunteers: they were not given any remuneration (financial or otherwise). Fig. 9 shows the correspondence between the distribution of layout appropriateness in terms of percentage (represented as a divergent horizontal stacked bar based on a Likert R package - http://jason.bryer.org/likert/) and the distribution of ranking (represented as vertical bars). The coding scheme is as follows: red=strongly disagree, orange=disagree, yellow= neutral, light green=agree, dark green= strongly agree). The yellow neutral part is divided into two equal parts in Fig. 9.The preferred variables for layout appropriateness are respectively: shape (82% of the 30 participants), value (71%), size (66%), motion (61%), color (46%), orientation (40%), and texture (22%). Shape was also ranked first followed by motion. Value was ranked second the most frequently, followed by shape. Motion was ranked third the most frequently, followed by shape. With seventh rank most frequently assigned, texture was judged the least appropriate variable on which we can play for conveying adaptivity because of its questionable legibility. For these reasons, shape was selected as the variable to indicate adaptivity.

In order to fulfill (R1)-(R5) requirements, a circular cloud layout was finally adopted, where font size and distance to the center represent the importance of a menu item, but where distance between the items does not represent their similarity (such as in document semantic tagging [17]) or adjacency in the initial menu (although this option could be further investigated). The rationale behind the choice is justified by the following findings and motivations:

- Among all referents (Fig. 2-7), the most preferred word cloud was a 2D circular (for its distinctiveness with the rectangular are of the initial menu) shape with horizontal items (in order not to reduce item legibility), without any animation (because it is hard to control and lengthy [9]), without any color or font effect apart from font size.
- Circular layouts have additionally shown to be most effective to spot high frequency terms in word clouds [22]: layouts generated by these algorithms are compact and clear, reduce unused white space and may feature arbitrary convex polygons as boundaries (which is not really usable for menu, though). The results of the user study and the technical evaluation enable designers to devise a combination of algorithm and parameters which produces satisfying word cloud layouts for many application scenarios [28].
- Users prefer a less extensive menu hierarchy on a small screen device and that item category classification and item labelling influence item selection performance [27].
- Small screens do not display more than 15 items simultaneously as opposed to dozens on desktop, thus providing an opportunity for applying this finding: provide a direct access to menu items that are the most predicted, since there is no room enough for displaying them all [27].
- For a small list with emphasized items, ocular saccades should be easier to perform than on linear list [6,21].

Step 3: Final Implemented Design of Cloud Menus

The final design of cloud menus is developed as a widget in Java for Eclipse based on Android Software Development

Kit. The Cloud Menu consists of a linear list for the static menu superimposed by a prediction window materialized as a circular word cloud with three prediction levels (Fig. 1): (1) any item with high prediction (probability ≥ 80%) is located in the center of the circle highlighted with large font size; (2) any item with medium prediction (60% ≤ probability ≤ 80%) is presented in the periphery with a decreasing size font and a larger distance from the center depending on the probability (the lower the prediction, the more far and the smaller the item becomes) ; (3) any item with low prediction (probability ≤ 60%) is displayed only in the static menu.

The Cloud Menu for Findlater test is depicted in Fig. 1: the France item with high prediction is presented in the center with the largest font size; Venus and Pecan with medium prediction are presented afterwards on a position indicating the original position of item in the static menu; Mercury, Walnut and Saturn with low prediction are located in the periphery with an increasing distance from the center and a decreasing size font. Note that the Pecan item is located on an imaginary line indicating an off-screen location. When prediction is correct, the user selects the item directly from the Cloud Menu. When prediction is incorrect, the user makes the cloud menu disappearing by pressing the ⊟ button and selects the item from the static menu after scrolling down/up.

CONTROLLED EXPERIMENT

A controlled experiment was conducted based on two conditions: a *Control condition* (or baseline), which presents the Findlater test in a static menu without any adaptation and prediction and a *Cloud Menu*, which presents the Findlater test as a Cloud Menu with 6 items in the prediction window. In order to test the influence of the circular layout of the Cloud Menu, a third condition was developed: the *Linear Menu*, where both prediction window and full list of items are presented as superimposed linear lists (Figure 4), but without any cloud. In order to backward and forward prediction window, the two [–] and [+] pushbuttons are also added.

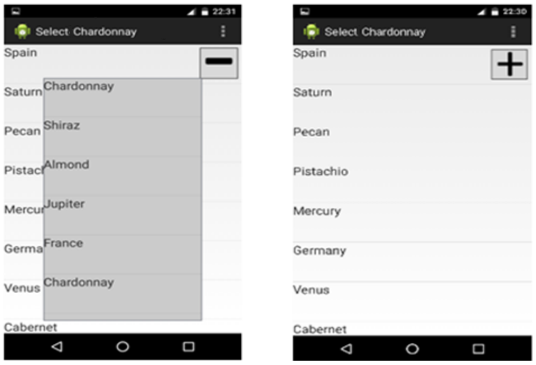

Figure 10. Linear Menu: a linear list with 6 predicted items.

Hypotheses. We made the following assumptions:

Speed with high prediction

H_{11}=The Cloud Menu and the Linear Menu will be faster than Control condition. When prediction is correct, the user finds the target item among 6 predicted items in the Cloud Menu and in the Linear Menu more quickly than in Control condition where the target belongs to a list of 16 items.

H_{21}=The Cloud Menu will be faster than the Linear Menu. The circular layout of the 6 predicted items with different font sizes and positions is faster than a linear list without any visual distinction between predicted items, as said in [29].

H_{31}=A target item located in the center of the Cloud Menu will be selected faster than when located in its periphery. Indeed, a target item located in the center is considered easy to find because it is emphasized with a large font size (thus inducing a larger activation area as in AAMU [33]), contrarily to a target item located in another part of the cloud menu.

Speed with low prediction

H_{41}=The Cloud Menu and the Linear Menu will not be worse than Control condition. Cloud and Linear adaptive menus will not be worse than the static menu as they avoid penalizing interaction when prediction is incorrect. The ⊟ button should enable the end user to escape from the Cloud Menu and Linear menu in case of an incorrect prediction.

H_{51}=The Cloud Menus will be faster than the Linear Menu. Exploration of predicted items will be easier in the Cloud Menu, where items are highlighted with different font sizes and positions, than in the Linear Menu.

H_{61}=In all conditions (Cloud Menu, Linear menu and Control condition), the target item will be selected faster when located on the first interaction screen than on the second. Accessing a target item on the first screen will be faster and easier than on the second screen requiring vertical scrolling.

Error rate with high prediction

$H_{71=}$The error rate of the Cloud Menu is similar to the Linear Menu. When the target item is one of the 6 predicted items in Cloud and Linear Menu, the visual search time and selection time are reduced, which also reduces error rate.

$H_{81=}$When target item is located in the center of the cloud, errors will be less frequent than when target is in its periphery. User attention will be attracted to the center where an item is displayed with a larger font size compared to items in other parts of the cloud, thus facilitating the item selection.

Error rate with low prediction

$H_{91=}$No significant difference between all conditions. When prediction is incorrect in both adaptive conditions (Cloud and Linear), the prediction window will not be used and will disappear with ⊟, thus generating the same error rate.

Methodology

There are three independent variables: (1) the MENU TYPE: Control condition, Cloud Menu, and Linear Menu, (2) the TARGET LOCATION: in the center, in its periphery, or outside, (3) the LEVEL OF PREDICTION: low, high. Two levels of prediction were decided to test the performance when prediction algorithm works well and bad, they were selected randomly. Each menu is divided into two screens of 8 items, all coming from Findlater test [10]. Predicted items appearing in the Cloud Menu and Linear Menu are controlled. The target item can be located either on the prediction window or on the static menu and its distribution is controlled. The target item in Cloud condition was controlled randomly between three locations: in the center, in its periphery, or outside in case of incorrect prediction. Accurate prediction level is when prediction is correct and target item is inside the Cloud Menu without any restriction. Inaccurate prediction level is when prediction is incorrect and target item is outside the Cloud Menu. The same behavior occurs in the Linear Menu where the target item was also controlled randomly between two prediction accuracy levels. High prediction level is when prediction is correct and target is one of the six predicted items presented on the prediction window. Low prediction level is when prediction is wrong and target is outside the prediction window. For both high and low prediction levels, the target item always appears in the static menu.

Task

A between-subjects design was decided to avoid any carryover effects such as practice (learning effect) and fatigue: two independent groups of participants were asked to perform a sequence of item selections. Participants of the first group tested Cloud Menu and Control condition, while the second group tested Linear Menu and Control Condition. Participants were divided into two groups using *matched-group design*, through which the subjects were matched according to their age and then allocated into group. For each condition, first, a message appeared indicating the target item to select. Then a list of items appeared and the target item was displayed at the screen top as a reminder. In Cloud Menu, participants selected the target item from the Cloud of predicted items and/or from the static menu. In Linear Menu, participants selected the target item from the prediction window and/or full list of items. In Control condition, participants selected target items from the main list on the first or on the next screen. When the selected item matched the requested target item, a new message appeared indicating next target item until the test was complete. When the selected item did not match the requested target item, an error message prompted the participant to select the right target before moving to the next target. When the test was complete, a “thank you” message was displayed.

Selections were performed by finger touch on the touchable surface of the smartphone, no stylus or pen were used. Generally, participants were holding the smartphone in their left hand and had to point with right hand (index finger). In each menu, order and position of items were controlled and changed randomly after ten selections in order to avoid learning effect. Selection sequence (target selection) was also randomly controlled. Target position on first screen or on second screen and prediction accuracy level were also controlled. Each participant had to execute 20 item selections in the Control condition, 20 item selections in the Cloud+High prediction condition and 20 item selections in the Cloud+ Low prediction condition, 20 item selections in the Linear +High prediction condition and 20 items selections in the Linear+Low prediction condition. Selections in the Cloud and Linear conditions were mixed in order to avoid any learning effect induced by a repetitive usage of a task.

Quantitative and Qualitative Measures. The dependent variables measured were: 1) menu item selection time (in seconds), which was measured as the time taken from opening the menu until final selection of requested target, and 2) error rate (in percentage %).

Apparatus. Android-based Google Nexus smartphones were used, with 2 Gb LPDDR3 RAM, 16 Gb of storage and a 1920 x 1080 pixel screen resolution (423 ppi).

Participants. Two independent groups of nineteen subjects each participated in this experiment. All participants were regular smartphone users and they were recruited in our organization through a mailing list.

Procedure. Before starting the test, the principle of each condition (Cloud Menu, Linear Menu and Control) was explained to participants but prediction levels were not. Each participant trained with a pre-test composed of ten targets. A different item list was used in the pre-test than the one used in the test. Both groups selected 60 target items as follows:

Group 1: **40** target items for Cloud Menu (20 with high prediction and 20 with low prediction) = 20 items when prediction is correct (10 located in the center of the Cloud Menu and 10 when target is in the periphery) + 20 items when prediction is wrong (items are all outside the Cloud Menu: 10 located on the first screen and 10 on the second screen). There were also **20** items for Control condition: 10 items located on the first screen and 10 on the second screen.

Group 2: **40** targets for Linear Menu = 20 items when prediction is correct and 20 items when prediction is wrong (target is outside the prediction window: 10 located on the first screen and 10 on the second screen). Similarly to group 1, there were also **20** items for Control condition: 10 items located on the first screen and 10 on the second screen.

In summary, the design was as follows:
19 participants ×
2 groups ×
60 targets ×
= 2280 menu item selections in total.

Group	Menu	Selection time (sec.)			Error rate (%)		
		μ	σ	W/Z	μ	σ	W/Z
G1 G2	Control Control	3.40 3.04	1.07 1.07	0.93	0.17 0.33	0.29 0.62	0.22

G1	Cloud (P+)	1.76	0.61	3.08	0.95	1.20	1.47
G2	Linear (P+)	4.12	3.08	***	1.93	2.12	
G1	Cloud (P+)	1.76	0.61	4.11	0.95	1.20	3.04
G1	Control	3.40	1.07	***	0.17	0.29	**
G1	Cloud (P-)	5.60	1.65	3.08	0.74	1.74	1.47
G2	Linear (P-)	3.40	2.84	**	2.66	2.79	
G1	Cloud (P-)	5.60	1.65	4.17	0.74	1.74	1.22
G1	Control	3.40	1.07	***	0.17	0.29	
G2	Linear (P+)	4.12	3.08	2.04	1.20	2.11	2.54
G2	Control	3.04	1.07	*	0.33	0.62	*
G2	Linear (P-)	3.40	2.84	0.20	2.66	2.79	3.54
G2	Control	3.04	1.07		0.33	0.62	**

Table 2. Results: P+=high prediction, P–=low prediction – No significance= *p*>.05, *= *p*≤ .05, **= *p*≤ .01, *=*p*≤ .001**

Results and Discussion

Levene's test was applied to verify homogeneity of variance between the two independent samples. Since this later was partially determined, non-parametric Mann-Whitney Comparison test was used for analysis between independent samples (groups), and Wilcoxon Signed Ranks test was applied in the case of within-subjects conditions (Table 2). Data were submitted to a Bonferroni Type I correction before handling.

Selection time

Selection time for all conditions is reported in the third column of Table 2 and graphically depicted in Fig. 11 with a 95% confidence interval for difference between normal means (α=.05). First of all, according to **row 1** in Table 2, there is no significant difference (Z=.93, *p*=.35) in Control condition between group G1 who tested the cloud menu (μ=3.40, σ=1.07) and group G2 (μ=3.04, σ=1.07), which allows us to properly compare these two independent groups.

According to **row 3**, Cloud condition with <u>high</u> prediction (P+: μ=1.76, σ=0.61) is significantly faster (W(22)=4.11, *p*= .00004) than Control condition (μ=3.40, σ=1.07). Similarly, Cloud condition with <u>high</u> prediction (P+: μ=1.76, σ=0.61) is significantly faster (Z=3.08, *p*=.002) than Linear condition in both cases as indicated in **row 2**: when target is in the center of the Cloud (μ=1.19, σ=0.54) and when it is in the periphery (μ=2.35, σ=1.04). Interestingly, when target is located in the center of the Cloud, participants are also significantly faster (W(23)=3.71, *p*=.0002) than when it is located in the periphery. Usually, corner locations are faster to reach.

Row 6 suggests that Control condition (μ=3.04, σ=1.07) is faster (W(14)=2.04, *p*=.05) than Linear condition when prediction is <u>high</u> (P+: μ=4.12, σ=3.08). In addition in **row 4**, when prediction is <u>low</u>, users are significantly faster (Z=3.08, *p*=.002) in Linear condition (P–: μ=3.40, σ=2.84) than in Cloud condition (P–: μ=5.60, σ= 1.65). In **row 5**, we observe that users are also significantly faster (W(22)=4.17, *p*=.0003) in Control condition (μ=3.40, σ= 1.07) than in Cloud condition with <u>low</u> prediction (P–: μ=5.60, σ=1.65).

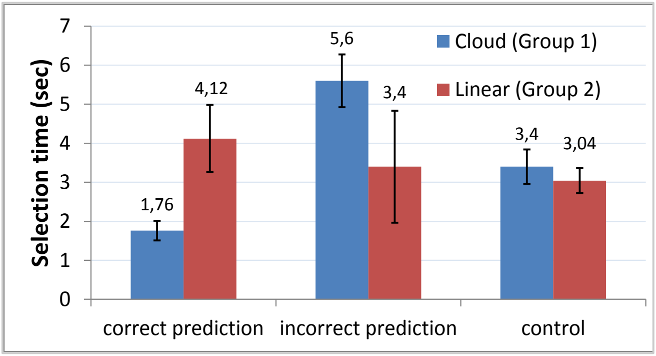

Figure 11. Selection time for all conditions (normal mean, α=.05).

More detailed results further suggest that:

1. Cloud with low prediction and target on <u>first</u> screen of the main list (P–: μ=4.98, σ=1.76) is significantly faster (W(22)=4.14, *p*=.00003) than Control with target on first screen (μ=2.42, σ=0.83)
2. Cloud with low prediction and target on <u>second</u> screen of the main list (μ=6.24, σ=1.63) is significantly faster (W(22)=3.56, *p*=.0003) than Control with target on second screen (μ=4.40, σ=1.78).
3. In Control condition, users are also significantly faster (W(22)=4.14, *p*=.00003) when the target is on the first screen (μ=2.42, σ=0.83) than when it is located on the second screen (μ=4.40, σ=1.78).

However, **row 7** reveals that there is no significant difference (W(14)=0.20, *p*=.84) between Control condition (μ=3.04, σ=1.07) and Linear condition when prediction is <u>low</u> (P–: μ=3.40, σ=2.84). More detailed results further suggest the following absences of significance:

1. No significance (W(14)=0.91, *p*=.36) between Linear with <u>low</u> prediction and target on first screen of the main list (P–: μ=3.55, σ=3.42), Control with target on first screen (μ=2.11, σ=0.53).
2. No significance (W(14)=1.14, *p*=.25) between Linear with <u>low</u> prediction and target on second screen of the main list (μ=3.25, σ=2.35) and Control with target on second screen (μ=3.97, σ=0.86).

However, users belonging to the Control condition are significantly faster (W(14)=3.40, *p*=.0006) when the target is in the first screen (μ=2.11, σ=0.53) than when it is located on the second screen (μ=3.97, σ=0.86), which is of course normal since any navigation to the second screen will inevitably increase the selection time. Fig. 12 depicts the item selection time by participant in the two groups by condition.

***H_{11}* is supported**. When prediction is high, end users find the target item among the 6 predicted items in the Cloud Menu and the Linear Menu faster than in Control condition when the target belongs to a list of 16 items, which somewhat normal since item selection is operated with a smaller amount of time included in a smaller surface, which is in line with [30].

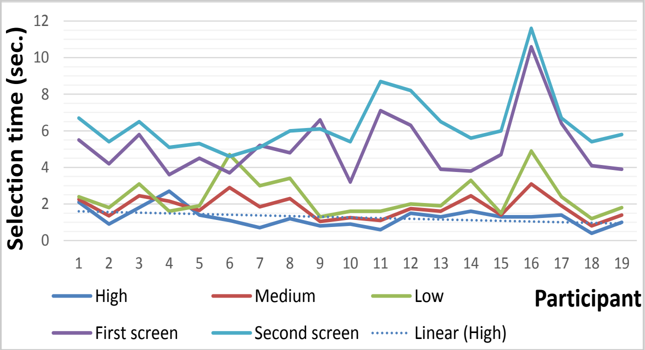

Figure 12. Selection time by participant.

***H_{21}* is supported.** When prediction is high, users are faster in Cloud menu than in Linear condition. This result can be justified by the fact that circular form of the Cloud facilitates exploration of a number of predicted items assigned to 6, which is higher than the typical 3 items found in other Split Adaptive menus. The user attention is distributed to different sides of the Cloud in contrary to the Linear condition when the user has to browse all items one by one. Consequently, by transitivity, Cloud Menu is faster than Linear menu, which is in turn faster than Control condition, which is the most important conclusion that demonstrates the effectiveness and efficiency of Cloud Menus.

***H_{31}* is supported**. This is justified by the fact that user attention is often attracted by the item in the center of the Cloud Menu because it is consistently located and it is always emphasized by a larger font size compared to other predicted items in the Cloud, which is consistent with [22]. Therefore, the central item is always the most pre-eminent among all menu items and its position is very predictable, even it requires some scanning. Suggested conclusions are:

1) Prediction step is crucial, it reduces navigation time and visual search time especially when menus are long.
2) Increasing the amount of predicted items is not penalizing the interaction because it is still better than the non-adaptive menu provided that this amount is not prohibitive, which may require another study to determine the threshold beyond which the positive effect no longer occurs.
3) The presentation of the prediction window as a circular word cloud may be an important factor making the Cloud Menu efficient. The results suggest that the Cloud Menu is better than the Linear Menu because of its impact on user perception [21] and usability of word clouds [22,32].
4) Linear menu is competitive to scan items (one eye fixation, one column), but becomes surpassed by a cloud menu (many eye fixations, multiple directions) when too many items increase the length of the split area.

***H_{41} and H_{51}* are partially supported.** Results showed that, overall when prediction is low, Control condition is faster than Cloud Menu and no significant difference is detected between the Linear and Control conditions. This result can be justified by the fact that being sure that the target is not one of the predicted items is easiest in the case of a linear list than in the Cloud. In this latter, the human perception is going in all directions and the user scans the Cloud several times in order not to miss any item. In Linear Menu, the user is often sure not to miss the target item since the whole menu is browsed sequentially. But as in the Cloud Menu, six predicted items are conveyed to the end user, thus implying that there is a higher probability to present the item of interest to the end user than when three predicted items are presented.

***H_{61}* is supported**. Users can have direct access to target when this latter is on first screen contrary to when it is on the second screen, in this case scrolling is required that can justify the fact that selection time is shorter on first screen than on second screen.

Error Rate

Error rate for all conditions is reported in the fourth large column of Table 2 and graphically depicted in Fig. 13 with a 95% confidence interval for difference between normal means (α=.05). Error rates are assessed as equivalent (Z= 0.22, *p*=.82) in Control condition both in G1 (μ=0.17, σ= 0.29) and G2 (μ=0.33, σ=0.62) as reported in **row 1** of Table 2. Overall, there is no significant difference observed (Z=1.47, *p*=.14) in terms of errors between the Cloud condition (μ=0.95, σ=1.20) and Linear condition (μ=1.93, σ=2.12) as indicated in **row 2**. In Cloud Menu, when prediction is high and item target is located in the center of the Cloud (P+: μ=0.04, σ=0.21), errors are less frequent (W(22)=3.44, *p*=.0005) than when the target is located in the periphery (μ=1.87, σ=2.40). In **row 3**, we observe that errors are significantly less frequent (W(22)=3.04, *p*=.002) in Control condition (μ=0.17, σ= 0.29) than in Cloud Menu (μ=0.96, σ=1.20), which may be due to tally errors in the periphery of the Cloud. When prediction is low in Cloud Menu (P–: μ=0.74, σ=1.74), there is no significant difference (W(22)= 1.22, *p*=.22) with respect to the Control condition (μ=0.17, σ=0.29) as reported in **row 5**. Similarly, errors are less frequent (W(14)=2.54, *p*=.01) in Control condition (μ=0.33, σ=0.62) than in Linear condition (μ=1.20, σ=2.11) as reported in **row 6** when prediction is high. When prediction is low in **row 7**, it is even more significant: errors are less frequent (W(14)=3.54; *p*=.007) in Control Condition than in Linear case with low prediction.

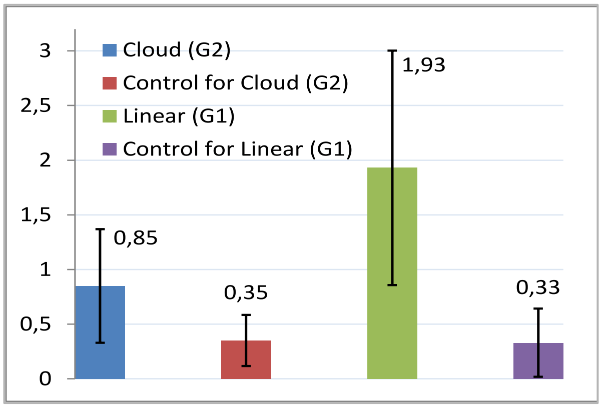

Figure 13. Error rate for all conditions.

***H_{71}* is supported**. When target is one of the 6 predicted items in Cloud and Linear Menu, the visual search time and selection time are reduced which reduce errors number.

***H_{81}* is supported**. In the periphery, some items can be close together, which suggests that a large number of tally errors can be generated. Contrarily to a target located in the center, locations, directions, and font size are factors minimizing tally errors. Items predicted inside the cloud can be laid out in a more optimized way by calculating the distance between items and the center, and distributing them, e.g., by considering their semantic relation. Further investigation is required to improve this aspect, by laying these items out depending on their position in the static menu.

***H_{91}* is supported**. In the case of low prediction, participants did not rely on the Cloud Menu and used the ⊟ button to make it disappear. The target is in the static menu, like in Control condition, which suggests that there is no difference between Cloud Menu and Control condition.

Experiment overview

The experiment conducted on the cloud menu corroborates several findings from Lohmann *et al.* [22]:

- *Item font size*: items with a large font size attract more user attention than with a small font size (an effect influenced by other parameters, such as item length, item position, and item neighboring). According to this study, recognition for items with a larger font size was significantly higher than items with a smaller font size: 83%, 73%, and 59% respectively for the three first font sizes. This also confirms the effect observed in Adaptive Activation-Area Menus [33], where a large selection area attracts more user attention than with a small area. AAMUs are spatially and physically instable, but preserve format and temporal stabilities. Conversely, Cloud Menus leave untouched the initial (static) menu.
- *Scanning*: cloud menus have been proved as an efficient adaptive split menu for small screens because participants tend to scan menu items rather than reading them, which accelerates their processing time. Concentrating this scanning into a designated area fosters this focus as opposed to distribute predicted items or to move the split area into another location, which is feasible for desktop, but not for small screens.
- *Centering*: menu items located in the middle of the cloud attract more user attention than tags near the borders, an effect influenced by radial layout like in pie menus. *H_{31}* and *H_{81}* are two supported hypotheses that confirm this finding.
- *Position*: the upper left quadrant receives more user attention than the others, but we did not exploit this effect since menu items are positioned so that they can point to their original position in the static menu, even when they are off-screen. This facility is optional and we did not identify any significant difference between a location-dependent item positioning and a location-independent one.

DESIGN GUIDELINES FOR CLOUD MENUS

A cloud menu represents a split adaptive menu for which the following design guidelines can be devised.

G1. A cloud menu should be used for a substantive static menu. The main idea behind using a tag cloud in information retrieval is that the relevance of a document must be determined with respect to a set of documents before this document actually appears. Similarly, the main idea behind using a tag cloud as an adaptive split menu is that the probability of selecting a menu item among other items in that menu should appear before the menu is entirely browsed and displayed. It does not make sense to produce an adaptive split menu such as the cloud menu for an initial (static) menu containing only a few items, even on a small screen. In our experiment, the half of the menu is visible: 8 items among 16.

G2. A cloud menu should not hold more than 6 items. So far, split adaptive menus have been mainly explored for large screens (e.g., laptop, desktop, large monitors, wall screens). Even under these conditions, 3 to 4 items were recommended [9,10] as the maximum threshold. The results of the experiment conducted suggest that this threshold could be upgraded up to 6, even on a small screens. Another study could conduct the same experiment on a large screen to determine to what extent the benefit is more important.

G3. A cloud menu should not exceed 3 levels of prediction. Beyond this threshold, the end user is likely to be no longer able to make any difference between the three levels. This is somewhat consistent with the 3 levels of highlighting recommended in usability guidelines. Other coding schemes could augment this representation, but may also increase the cognitive load as opposed to reinforcing the same data.

G4. A cloud menu should be located as close as possible to the static original menu. When a split adaptive menu is located too far away from its original menu, there is a risk of losing the semantic or physical relationship between the static and the predicted parts. This is consistent with the recommendation from [29,30] to minimize visual displacement between the various regions.

G5. The items of a cloud menu should be located as much as possible to point to their corresponding (static) items. When both are on the same display, they should be positioned on the same line (plain red arrow in Fig. 1). When a predicted item refers to an off-screen location, it should also be positioned to indicate this situation (dotted red arrow in Fig. 1). This guideline is applicable to any adaptive split menu, but is even more important for desktop applications.

G6. A cloud menu on a small screen can be superimposed. While large screens can accommodate another (close) location for displaying the prediction window, a small screen is unable to satisfy the same constraint. Thus, a superposition avoids creating another parallel menu like in the traditional split menu, with the risk of oscillating between the two. It also preserves spatial, physical, and format stability (since the static menu is left untouched), but not temporal stability.

G7. A cloud menu should optimize its circular layout. Since shape was elicited in the focus group as the first variable to manipulate for materializing a cloud, other parameters, like color, texture, animation should be left out. Instead, the circular layout could be optimized based on [29], with only one item per line and only one background and foreground color. We did not play with transparency like alpha blending.

CONCLUSION AND FUTURE WORK

This paper presented Cloud Menus, a new type of adaptive split menu in which predicted menu items are arranged in a circular word cloud superimposed on the static menu. Menu items adhere to a series of seven design guidelines that have been devised. Through a controlled study, the cloud menu has been shown as a promising interaction technique for accelerating interaction with menus on small screens (e.g., watches, mobile phones, smartphones): due to its adaptivity, it exhibits a faster and more reliable behavior than a static (non-adaptive) menu, whether the prediction is correct or not. The circular layout was demonstrated to be superior to the Linear Menu, in which the adaptivity is identical, but the prediction window is presented as a linear menu instead.

Although tag clouds and word clouds (as used in this adaptive split menu) are becoming more popular and ubiquitous, especially after they have been offered as free widgets in many environments like jQuery, WordPress, Joomla!, and even Unity, we only know a little about usability of word clouds, apart from [22]. This paper contributes to a better understanding of their usability when applied as a cloud menu, but much remains to be done to examine its full potential. 3D clouds are also becoming popular and, yet, their usability is even less known. Their usability is likely to be largely affected by the visibility of the items which are subject to animation without any user control. In our preliminary experiment, motion came in the fourth position, but only for a 2D cloud menu. Studying a 3D cloud menu is another thing: items are floating around, thus generating occlusion.

In principle, the font size of a word in a word cloud is determined by its incidence. For smaller frequencies one can specify font sizes directly, from one to whatever the maximum font size. For larger values, a scaling should be made. Since the amount of predicted items is limited, it does not make sense to devote computational time for calculating the font size dynamically with respect to all levels of prediction.

In future work, we will investigate the impact of the cloud layout on menu selection, namely based on alternate prototypes (Fig. 2) so as to better study the distance of an item to the center of the Cloud as a function of accuracy level of this item. Also studying and improving presentation of predicted items allows us to minimize the number of errors, namely by a systematic analysis of Bertin's visual variables, either for one visual variable at a time or for a combination of them. Throughout this paper, we hope to open and encourage exploring other directions in split adaptive interface. We hope that both researchers and practitioners will benefit from this new form of split adaptive menu.

REFERENCES

[1] L. Antwarg, T. Lavie, L. Rokach, B. Shapira, J. Meyer. 2013. Highlighting items as means of adaptive assistance. *Behaviour & Information Technology*, 32, 8, 761–777. http://dx.doi.org/10.1080/0144929X.2011.650710

[2] S. Asthana, P. Singh, A. Singh. 2013. Exploring adverse effects of adaptive voice menu. In *Proc. of CHI EA'2013*, 775–780. https://doi.org/10.1145/2468356. 2468494

[3] G. Bailly, E. Lecolinet, L. Nigay. 2016. Visual Menu Techniques. *ACM Computing Surveys* 49, 4, article #40. http://dx.doi.org/10.1145/3002171

[4] J. Bertin, *Sémiologie graphique*. Paris, Mouton/Gauthier-Villars, 1967.

[5] S. Bouzit, G. Calvary, D. Chêne, J. Vanderdonckt. 2016. A design space for engineering graphical adaptive menus. In *Proc. of EICS'2016*, 239–244. https://doi.org/10.1145/2933242.2935874

[6] R. Bridle, E. McCreath. 2005. Predictive Menu Selection on a Mobile Phone. In *Proc. of Workshop W7 on mining spatio-temporal data* (ECML/PKDD'2005). Retrieved Aug. 24th, 2017 from https://cs.anu.edu.au/~Eric.McCreath/papers/mining05.pdf

[7] R. Bridle, E. McCreath. 2006. Inducing shortcuts on a mobile phone interface. In *Proc. of IUI'2006*, 327–329. https://doi.org/10.1145/1111449.1111526

[8] J. Feinberg. 2010. Wordle. In *Beautiful Visualization:*

Looking at Data through the Eyes of Experts, J. Steele, N. Iliinsky (eds.). O'Reilly, Sebastopol. 37–58.

[9] L. Findlater, J. McGrenere. 2004. A comparison of static, adaptive and adaptable menus. In *Proc. of CHI'2004*, 89–96. http://dx.doi.org/10.1145/985692.985704

[10] L. Findlater, K. Moffatt, J. McGrenere, J. Dawson. 2009. Ephemeral Adaptation: The Use of Gradual Onset to Improve Menu Selection Performance. In *Proc. of CHI'2009*, 1655–1664. http://dx.doi.org/10.1145/1518701.1518956

[11] L. Findlater, K.Z. Gajos. 2009. Design Space and Evaluation Challenges of Adaptive Graphical User Interfaces. *AI Magazine* 30, 4, 68–73. https://doi.org/10. 1609/aimag.v30i4.2268

[12] Y. Fukazawa, M. Hara, M. Onogi, H. Ueno. 2009. Automatic mobile menu customization based on user operation history. In *Proc. of MobileHCI'2009*. Article #50. https://doi.org/10.1145/1613858.1613921

[13] K.Z. Gajos, M. Czerwinski, D.S. Tan, D. Weld. 2006. Exploring the Design Space for Adaptive Graphical User Interfaces. In *Proc. of AVI'2006*, 201–208. http://dx.doi.org/10.1145/1133265.1133306

[14] K.Z. Gajos, K. Everitt, D.S. Tan, M. Czerwinski, D.S. Weld. 2008. Predictability and accuracy in adaptive user interfaces. In *Proc. of CHI'2008*, 1271–127. https://doi.org/10.1145/1357054.1357252

[15] K.Z. Gajos, K. Chauncey. 2017. The Influence of Personality Traits and Cognitive Load on the Use of Adaptive User Interfaces. In *Proc. of IUI'2017*, 301–306. https://doi.org/10.1145/3025171.3025192

[16] M. Halvey, M.T. Keane (2007). An Assessment of Tag Presentation Techniques. In *Proc. of WWW'2007*, 1313–1314. http://dx.doi.org/10.1145/1242572.1242826

[17] F. Heimerl, S. Lohmann, S. Lange, T. Ertl. 2014. Word Cloud Explorer: Text Analytics based on Word Clouds. In *Proc. of HICSS'2014*, 1833–1842. http://dx.doi.org/10.1109/HICSS.2014.231

[18] S.C. Huang, I-F. Chou, R.G. Bias. 2006. Empirical evaluation of a popular cellular phone's menu system: theory meets practice. *Journal of Usability Studies* 1, 2, 91–108. http://uxpajournal.org/empirical-evaluation-of-a-popular-cellular-phones-menu-system-theory-meets-practice/

[19] T. Kühme, U. Malinowski, J.D. Foley. 1993. Facilitating interactive tool selection by adaptive prompting. In *Proc. of InterCHI'93*, 149-150. https://doi.org/10. 1145/259964.260160

[20] T. Lavie, J. Meyer. 2010. Benefits and costs of adaptive user interfaces. *Int. J. of Human-Comp. Studies* 68, 8, 508–524. http://dx.doi.org/10.1016/j.ijhcs.2010.01. 004

[21] W. Liu, G. Bailly, A. Howes. 2017. Effects of Frequency Distribution on Linear Menu Performance. In *Proc. of CHI'2017*, 1307–1312. https://doi.org/10. 1145/3025453.3025707

[22] S. Lohmann, J. Ziegler, L. Tetzlaff. 2009. Comparison of Tag Cloud Layouts: Task-Related Performance and Visual Exploration. In *Proc. of INTERACT'2009*, 392–404. http://dx.doi.org/10.1007/978-3-642-03655-2_43

[23] L. Norrie, R. Murray-Smith. July 2016. Investigating UI Displacements in an Adaptive Mobile Homescreen. *Int. J. of Mobile Human Computer Interaction*, 8, 3, 1–17. https://doi.org/10.4018/IJMHCI.2016070101.oa

[24] J. Park, S.H. Han, Y.S. Park, Y. Cho. 2007. Usability of Adaptable and Adaptive Menus. In *Proc. of 2nd Int. Conf. on Usability and Internationalization* (UI-HCII'2007), LNCS 4559, 405–411. http://dx.doi.org/10.1007/978-3-540-73287-7_49

[25] M. Pielot, A. Kazakova, T. Hesselmann, W. Heuten, S. Boll. 2012. PocketMenu: nonvisual menus for touch screen devices. In *Proc. of MobileHCI'2012*, 327–330.

[26] A. Ponsard, K. Ardekani, K. Zhang, F. Ren, M. Negulescu, J. McGrenere. 2015. Twist and Pulse: Ephemeral Adaptation to Improve Icon Selection on Smartphones. In *Proc. of GI'2015*, 219–222. https://dl.acm.org/citation.cfm?id=2788929

[27] D. Raptis, N. Tselio, J. Kjeldskov, M.B. Skov. 2013. Does size matter?: investigating the impact of mobile phone screen size on users' perceived usability, effectiveness and efficiency. In *Proc. of MobileHCI'2013*, 137–136. http://dx.doi.org/10.1145/2493190.2493204

[28] A.W. Rivadeneira, D.M. Gruen, M.J. Muller, D.R. Millen. 2007. Getting Our Head in the Clouds: Toward Evaluation Studies of Tagclouds. In *Proc. of CHI'2007*, 995–998. http://dx.doi.org/10.1145/1240624.1240775

[29] K. Samp, S. Decker. 2011. Visual Search in Radial Menus. In *Proc. of Interact'2011*, 248–255. http://dx.doi.org/10.1007/978-3-642-23768-3_21

[30] A. Sears, B. Shneiderman. 1994. Split menus: Effectively using selection frequency to organize menus. *ACM Trans. on Comp.-Human Interaction* 1, 1, 27–51. http://dx.doi.org/10.1145/174630.174632

[31] C. Seifert, B. Kump, W. Kienreich, G. Granitzer, M. Granitzer. On the beauty and usability of tag clouds. In *Proc. of IV'2008*, 17–25. http://dx.doi.org/10.1109/IV.2008.89

[32] Ch. Shin, J.-H. Hong, A.K. Dey. 2012. Understanding and Prediction of Mobile Application Usage for Smart Phones. In *Proc. of UbiComp'2012*. 173–182. http://dx.doi.org/10.1145/2370216.2370243

[33] E. Tanvir, J. Cullen, P. Irani, A. Cockburn. 2008. AAMU: Adaptive Activation Area Menus for Improving Selection in Cascading Pull-Down Menus. In *Proc. of CHI'2008*. 1381–1384. http://dx.doi.org/10.1145/ 1357054.1357270

[34] T. Tsandilas, M.C. Schraefel. 2005. An empirical assessment of adaptation techniques. In *Proc. of CHI'2005 Extended Abstracts*, 2009–2012. http://dx.doi.org/10.1145/1056808.1057079

[35] T. Xie, T. Lin, Y. Zhu, R. Chen. 2015. Modeling Human Behavior to Reduce Navigation Time of Menu Items: Menu Item Prediction Based on Markov Chain. In *Human Behavior, Psychology, and Social Interaction in the Digital Era*. Chapter 8, IGI Global. http://dx.doi.org/10.4018/978-1-4666-8450-8.ch008