G-Menu: A Keyword-by-Gesture Based Dynamic Menu Interface for Smartphones

Jean Vanderdonckt[1(✉)] and Éric Petit[2]

[1] Université catholique de Louvain, 1348 Louvain-la-Neuve, Belgium
jean.vanderdonckt@uclouvain.be
[2] Orange Labs, 28 chemin du Vieux Chêne, 38240 Meylan, France
eric.petit@orange.com

Abstract. Instead of relying on graphical or vocal modalities for searching an item by keyword (called K-Menu), this paper presents the G-Menu exploiting gesture interaction and gesture recognition: when a user sketches a keyword by gesturing the first letters of its label, a menu with items related to the recognized letters is constructed dynamically and presented to the user for selection and auto-completion. The selection can be completed either gesturally by an appropriate gesture (called the G-Menu) or by touch only (called the T-Menu). This paper compares the three types of menu, i.e., by keyword, by gesture, and by touching, in a user study with twenty participants on their item selection time (for measuring task efficiency), their error rate (for measuring task effectiveness), and their subjective satisfaction (for measuring user satisfaction).

Keywords: Gesture interaction · Menu selection · Search by keyword

1 Introduction

As a mobile devices, such as smartphones and tablets, continue to embark more off-line functions and on-line services, their menu structure largely increases in terms of amount of menu items (e.g., labels in a list or icons in a palette), menu depth, and hierarchy. This menu immediately becomes constrained by the screen resolution of the target device, which is capable of only displaying a limited amount of items and a single menu level at a time. This forces the end user to scroll vertically to browse all items in a menu and to swipe left or right to navigate across menu levels. Therefore, instead of navigating through the menu structure, end users tend to search for an item by keyword, for example by typing progressively "Weather" to reach any application of weather forecasts. This technique, called a "K-Menu" [15], is a keyword text-entry menu (Fig. 1a). Existing operating systems usually offer the K-Menu for searching for a menu item by keyword: when a user enters a keyword, a menu with items related to the keyword is constructed dynamically by inspecting a table of menu items and presented to the user. The end user then selects an item in this list or exploits auto-completion to select the complete item when no ambiguity persists.
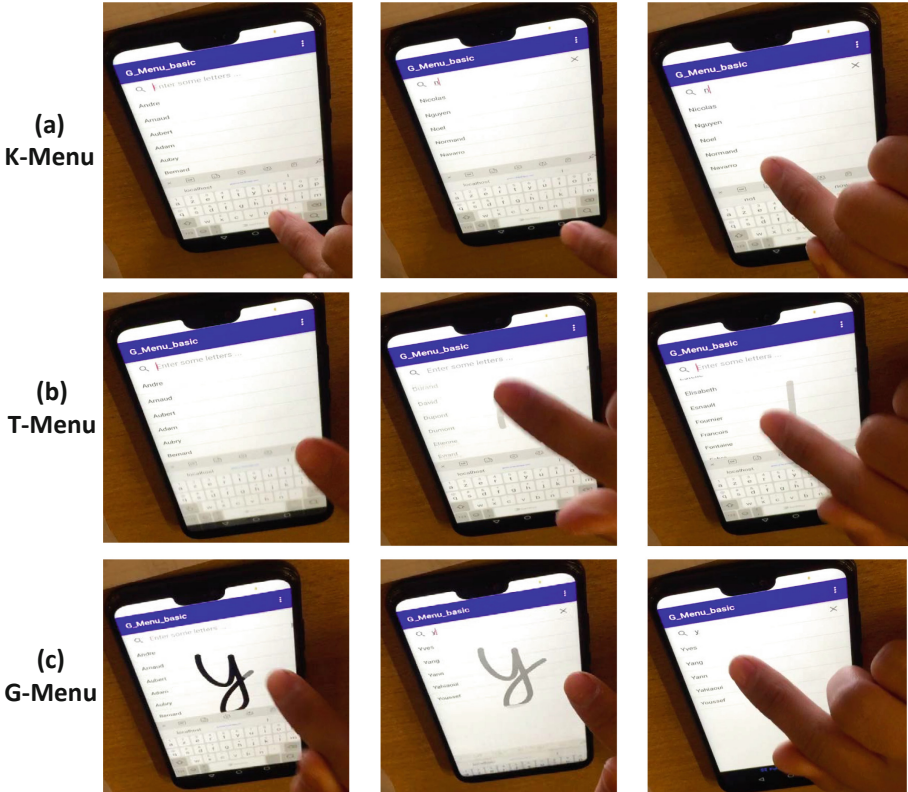
Fig. 1. The three menu conditions: (a) K-Menu for selecting the item "Navarro", (b) T-Menu for selecting the item "Fournier", and (c) G-Menu for selecting the item "Yang".

Operating systems also sometimes offer a similar capability supported by voice recognition: a pronounced word is captured vocally, then submitted to speech recognition, and transformed into a searchable keyword (keyword voice-entry menu). Often, no distinction is made between menu items and other labels used in settings of applications. The end user is still invited to select the desired items by touch, which comes back to the initial item selection mechanism.

Instead of relying on graphical or vocal modalities, this paper presents the *G-Menu* exploiting gestural interaction and recognition for keyword gesture-entry menu [26]: when a user sketches a keyword by gesturing the first letters of its label, a menu with items related to the recognized letters is constructed dynamically and presented to the user for further selection. The selection can be completed either gesturally by an appropriate gesture (called the *G-Menu*: Fig. 1c) or by touching (called the *T-Menu*: Fig. 1b). While the *K-menu* certainly remains the most frequently used and popular menu, the *G-Menu* and the *T-Menu* may offer new affordance that we did not thought of before. To better understand these differences, this paper compares the three types of menu, i.e.,

by keyword (*K-Menu*), by gesture (*G-Menu*), and by touching (*T-Menu*) in a user study with twenty participants on their item selection time (for measuring task efficiency), their error rate (for measuring task effectiveness), and their subjective satisfaction (for measuring user satisfaction).

The remainder of this paper is structured as follows: Sect. 2 discusses work related to the problem of selecting menu items on a smartphone by keyword and alternate approaches. Section 3 explains our implementation of the G-Menu and T-menu in the context of MenuDFa [20], an initiative to provide end users of mobile phones with an adaptive interface depending on their profile [6] as a Graphical Adaptive Menu (GAM) [25]. Section 4 defines three hypotheses to be investigated in an experiment to determine the potential advantages and shortcomings of a T-Menu and a G-Menu over the well-known K-Menu.

2   Related Work

This section is divided into two parts: prior work related to graphical adaptive menu (GAM) and selected work in this area to support menu adaptivity.

2.1   Graphical Adaptive Menus

Visual Menu techniques are not only very numerous, but also very diversified in terms of capabilities and implementations [2]. Graphical Adaptive Menus (GAMs) are a particular class of graphical menus where their menu items are subject to some adaptivity [11,25], i.e. some adaptation of the menu initiated by the system based on internal data of the end user, such as the navigation history [10]. Contrarily to menu recommendation which adapts the menu based on data external to the user (e.g., recommendations from a cluster of users, from the crowd), menu adaptivity only takes into account data internal to the user (e.g., navigation history, recency or frequency of usage [12]). Many forms of adaptivity may be considered [4], ranging from shorten the label of menu items [8] and changing the form of selection zone [1] to changing the input/output modality [27], such as in *polymodal menus* [5]. Menu adaptation, whether it is initiated and controlled by the system in adaptivity or by the end user in adaptability, could lead to different appreciations by end users [7], which in turn depend on individual traits of the end user [11]. Between full adaptability and adaptivity resides a wide range of mixed-initiative possibilities [16], which may be subject to optimization of menu selection [2]. To characterize this range more precisely, we revisit the Automation Level Description (ALD) [19], which defines ten levels of automation for any system where automation in general is defined as

> "the execution by a machine agent (usually a computer) of a function that was previously carried out by a human. What is considered automation will therefore change with time. When the reallocation of a function from human to machine is complete and permanent, then the function will tend to be seen simply as a machine operation, not as automation."

Based on this definition, we hereby define the Adaptation Automation as any component of the interactive application, primarily its Graphical User Interface (GUI), which achieves the GUI adaptation as a function that was previously ensured by the end user. Table 1 defines ten levels of Adaptation Automation, ranging from full adaptability (level = 1) to full adaptivity (level = 10). While this scale is useful for characterizing the Adaptation Automation Level (AAL), it requires further investigation on how to specify, design, and implement the functions involved in mixed-initiative as some of them are cumulative or exclusive.

Table 1. Adaptation Automation Level (AAL), revisited from [19].

Level	Description
1	The GUI offers no assistance: the end user must take all decision and actions
2	The GUI offers a complete set of decision/action alternatives, or
3	Narrows the selection down to a few, or
4	Suggests one alternative, and
5	Executes that suggestion if the end user approves, or
6	Allows the end user a restricted time to veto before automatic execution, or
7	Executes automatically, then necessarily informs the end user, and
8	Informs the end user only if asked, or
9	Informs the end user only if the GUI decides to
10	The GUI decides everything and acts autonomously, ignoring the end user

2.2 Interaction Techniques for Menu Adaptivity

FaThumb [14] is aimed at overcoming the limitations induced by keyword text-entry menu. Instead of typing a keyword, facet navigation across a hierarchy of metadata is promoted to differentiate items proposed: either a menu item is not relevant and it is de-emphasized, or it is relevant and it is highlighted based on iterative data filtering. Keyword text-entry menu is more powerful when the label of the target menu item is known, while facet navigation is more effective and preferred, especially when the label of the menu item researched in not known, but its characteristics revealed by facets are. TapGlance [21] uses a zooming metaphor to unify navigation within and across applications based on faceted search. Its spatial metaphor could therefore be considered as an alternative menu search for smartphones. Substituting the item selection based on graphical point-and-click paradigm by another modality, such as vocal or gestural, may present some interesting opportunities to investigate. Menu selection by gesture has already been investigated, mainly by directional gestures (i.e., a particular item is associated to the direction of a marking menu, which is easy to produce but not very meaningful), by pointing gestures (i.e., a particular item is contained in a dedicated region) or by combining directional gestures and letters (i.e., a particular item or set of items is assigned to a letter gesture followed by a marking menu in AugmentedLetters [22]). But these different gesture types may be subject to different recall rates or not [9]. For example,

M3 GESTURE MENU displays menu items on a grid of a smartphone, prefers gestural shapes rather than directional marks, and has constant and stationary space use.

3   Context of the G-Menu

3.1   Context and Motivations of the Study

Smartphones are probably the devices that are today used by the widest possible population when considering profiles, preferences, habits, and abilities of their impediment thereof. Towards this goal, Orange Lab releases and continually maintains *MenuDFa* [20], an Android OS-based package supporting user interface adaptation up to $AAL = 5$. Since several adaptation techniques could be considered to adapt the smartphone user interface to the end user, it is vital to retain only those techniques that have been empirically validated. Therefore, observing visually disabled and able-bodied users together through user studies allows to pinpoint several user interfaces elements important when building interfaces for sight-impaired people as well as visually-disabled ones.

3.2   Implementation

The *G-Menu* and *T-Menu* were developed in Java in Android Software Development Toolkit (SDK) based on the following modules:

– A stroke gesture recognizer captures and recognizes over 100 allographs, both unistroke and multistroke, covering the ten digits and twenty-six letters of the Latin alphabet both in lowercase and uppercase. For some cases, several variants of a same digit or figure are provided to support Anglo-Saxon style, such as for the 'seven' or the 'four' digits.
– A gesture recognition engine embedding the stroke gesture recognizer to avoid mismatching between menu-oriented gestures issued by the keyword and non-oriented gestures (e.g., navigation gestures such as swipes, flicks, and drags). This engine exploits dynamic attributes such as gesture velocity and execution time to distinguish direct pointing gestures (i.e., related to direct manipulation) from operative gestures (i.e., related to gesture command).
– A module constructing a dynamic menu based on the gesture recognition by keyword text-entry menu matching and menu ranking scheme.
– An interaction technique offering both gesturally and graphically-oriented menu selection with a log file storing item selection time and error rate.

The *G-Menu* relies on both the gesture recognition engine and the dynamic menu filtering with two gestural commands: the last letter input could be erased by a left flick and the complete entry could be erase by a left-right flick (round-trip). The G-Menu and the T-Menu are delivered in a package called "Tactile Facile"[1] with five predefined interaction profiles: `Easy+` mode which is the by

---

[1] https://boutique.orange.fr/informations/accessibilite-autonomie/tactile-facile.php.

default mode, `Vision+` mode for people having light visual disabilities, `Vision++` mode for people having important visual deficiencies, `Motor+` for motor-impaired users, and `MicroGesture` for exploiting micro-gestures.

3.3   Hypotheses

Since we are confronted with three types of menu, i.e., the *K-Menu*, the *G-Menu*, and the *T-Menu*, the main research question arises: which menu type is the best and under which conditions? Three variables are usually manipulated in an experiment to differentiate menus of these types: item selection time for measuring task efficiency, error rate for measuring task effectiveness, and subjective satisfaction for measuring user satisfaction. We therefore formulate three hypotheses:

H_{11} = *The users select items in the K-Menu faster than in the G-Menu and the T-Menu.* The goal of this first hypothesis is to verify that the K-Menu still remains the fastest menu for search by keyword. The keyword text-entry method has always been revealed the fastest one [14] since people are used to efficiently rely on a keyword.

H_{21} = *The users select items in a T-menu faster than the other menus if the target menu item is close to the location where users initiate the search.* Indeed, if the user has to find an item that is close to the current position, the selection will be faster by slightly scrolling up and down than by searching it with a keyword. The hierarchical structure of the application will make it faster for users to find words that are near to their position.

H_{31} = *The users produce more errors with a G-menu than with the K-Menu and the T-Menu.* The G-Menu is a new sort of menu in which drawing a letter is probably more difficult than just tapping a keyword on the screen or to scroll to a word. Therefore, we believe that the G-menu will produce more selection errors than the others.

4   Method

Procedure. Each participant performed the task in a controlled environment. Prior to the task each participant was welcomed, had the process explained to them, signed a consent form, and filled in a questionnaire on their background. After the questionnaire was completed, the experimenter demonstrated the three types of menu with one example of item selection each. The participants were given 5 min to familiarize themselves with the menus and ask any question. The participants could finish this part early, then received a list of items to select from (Fig. 2). They had to cover the whole list with a paper, then they had to find the first item with the corresponding menu category. Once they found the item using the specific menu, they were asked to select it to confirm, then they could move the paper to the second item, and so on until the list is completed. They were given 15 min to complete the task, which was assessed as much more than

needed. At the end, participants received a questionnaire and were interviewed to determine what they liked and what they did not like about each menu and about the experiment overall, in order to have a subjective feedback. Our study was within-subjects with one independent variable: the MENU TYPE, a nominal variable with three conditions, one representing the baseline (*K-Menu*) and two for testing (*T-Menu* and *G-Menu*).
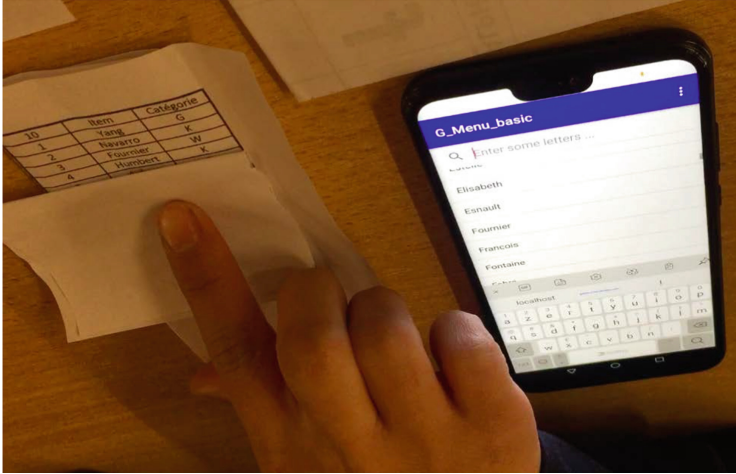

Fig. 2. Setup of the experiment with the list of 18 random items.

Stimuli. Thirty different lists of 18 items each (6 items for *K-Menu* + 6 items for the *T-Menu* + 6 items for the *G-Menu*) were randomly generated by using dCode (https://www.dcode.fr/tirage-au-sort-nombre-aleatoire) from a pool of 50 items extracted from the 130 menu items delivered in the MenuDfa application. All items were individually randomly presented and associated to one of the three conditions. The design was therefore as follows: 30 participants × 18 items = 540 samples. Each session was also video-recorded.

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

Quantitative and Qualitative Measures. The dependent variables were:

1. The menu item selection time (in sec), which was measured as the time taken from identifying the next requested target item in the list until its final selection.
2. The error rate (in percentage %), which was measured as the ratio of successfully achieved selections by the total amount of selections. Every time an item was not selected correctly, or every time the user had to go back in order to find the item, or every time a letter was drew wrong (for *G-Menu*), we counted it as one error with a maximum of 5.

3. The values filled in by each participant for a post-test questionnaire, which enables participants to express their level of satisfaction regarding five statements about the ease, the task completion, the speed, the learning and the productivity of each menu. Each statement is captured using a 3-point rating scale (1st = best, 2 = average, 3 = worst).

Analysis. After each participant completed the procedure, the measures, questionnaire, and ranking data was entered into a spreadsheet in an anonymous format so the participants could not be identified and be GDPR compliant.

5   Experiment

5.1   Participants

The sample included thirty participants (13 female and 17 male) recruited through mailing and contact lists, from different ages (min: 19, max: 70, $M = 25.73$, $SD = 10.05$), with diverse education degrees (i.e., secondary school, higher education, bachelor, master . . .). Although most of the participants were students in these different domains, there were also some other participants with various occupations (e.g., workers, unemployed, retired). In average, the participants were all well acquainted with the device (smartphone) and use it frequently (80%). However, older people were not necessarily as familiar with smartphones, which made them less comfortable with the device used, which influenced their data for the time and error rate. 40% of them never use a tablet, or almost never. On a frequency scale from 1 to 7, the rest of them vary between 5 and 2. No compensation was offered. Overall, the experiment lasted between 2 and 3 min per participant.

5.2   Results and Discussion

Out of the initial 540 trials, 33 outliers were removed for various reasons: they did not complete the full list (e.g., one item was skipped inadvertently), they selected an item for a wrong condition (e.g., G-Menu instead of K-Menu), the video record was interrupted, etc. The final breakdown was therefore: 170 *K-Menu* trials + 174 *T-Menu* trials + 170 *G-Menu* trials = 517 trials.

First Hypothesis: Selection Time. Figure 3 reproduces the item selection time aggregated for all participants, then for each participant. Not surprisingly, the K-Menu ($M = 4.09$, $SD = 2.32$) benefit from the fastest item selection time, followed by the T-Menu ($M = 8.07$, $SD = 5.30$), and the G-Menu ($M = 8.43$, $SD = 7.30$). All menus received a wide interval between the minimum and the maximum values: *K-Menu* ($min = 2, max = 18$), *T-Menu* ($min = 1, max = 35$), *G-Menu* ($min = 2, max = 50$). This is reflected that there is an important standard deviation between participants, the widest being for the *G-Menu*.
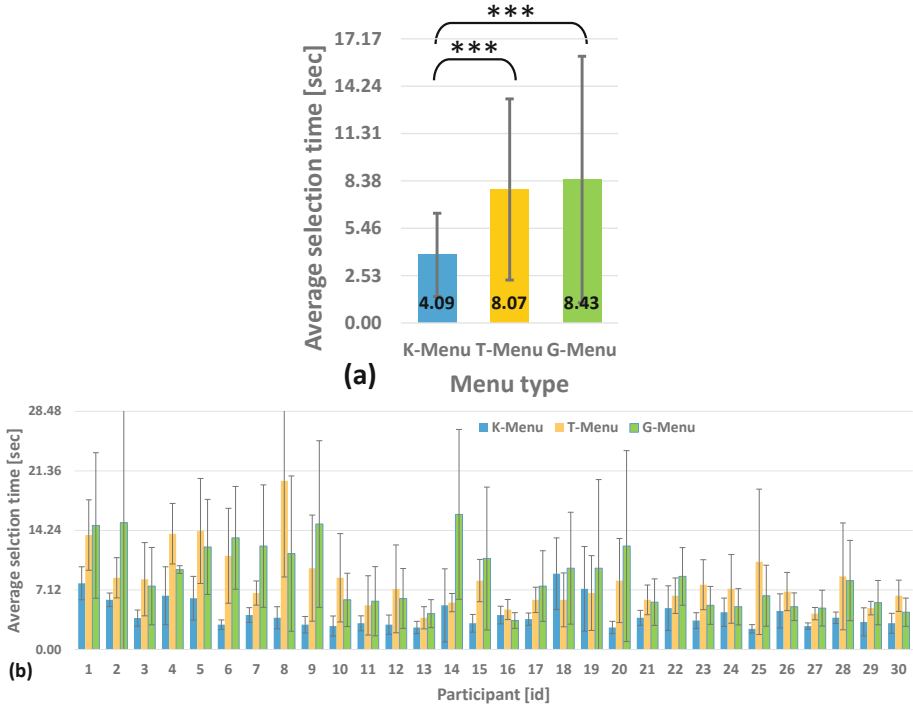
Fig. 3. Item selection time aggregated for all participants (a), then per participant (b). Error bars show the standard deviation.

We computed a series of Student's t-tests with two paired samples to determine whether there was any significant difference between the menu types. There was a very highly significant difference in the selection time for *K-Menu* and *T-Menu* conditions; $df = 342$, t = 1.64 for one-tail, t = 1.96 for two-tail, $p^{***} < .001$, Cohen's $d = .92$. There was also a very highly significant difference in the selection time for *K-Menu* and *G-Menu* conditions but with a smaller magnitude; $df = 341$, t = 1.65 for one-tail, t = 1.97 for two-tail $p^{***} < .001$, Cohen's $d = .81$. There was no significance between the *T-Menu* and *G-Menu* ($df = 345$, t = 1.65 for one-tail, t = 1.97 for two-tail, $n.s. = p > .05$). There is some concordance between those results and the statements assessed in the post-test questionnaire. Indeed, when asked to rank the three menu types regarding their speed, 20 participants out of 30 thought that the *K-menu* was the fastest to perform the task. Half of the participants thought the *G-menu* was the slowest, the other half thinking it was the *T-Menu* (Fig. 4-Speed). In conclusion, H_{11} is supported. **Second Hypothesis: Selection Time for *T-Menu* When Item is Close.** To test this hypothesis, we decided to work with two groups of items. First, it is the group of items whose location is close to the place where the participant initiates the search (relative displacement ≤ 41 [17]). When the participant is searching for keyword, she begins in general at the top of the menu. But when
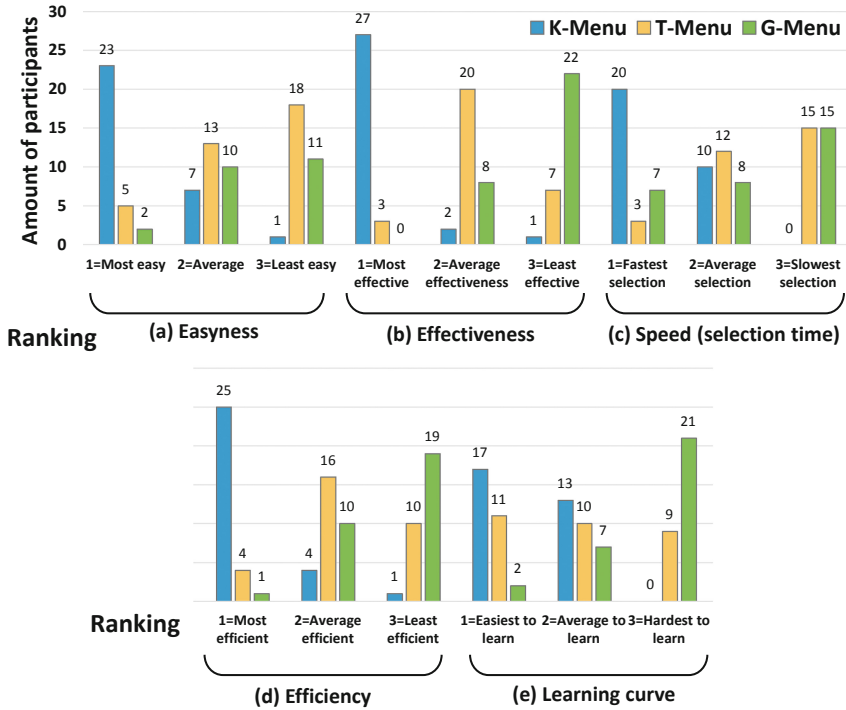
Fig. 4. Ranking of menu types per statement.

the participant utilizes T-Menu (mainly for scrolling) and that she selects the searched item, she stays at the same positions of this item in the menu. Therefore, we used the relative position instead of the normal position in the menu. This relative position, or relative displacement, gives us some information about how far the target item is from the participant's current locus of control. For an item with a relative position less or equal than 41, the *T-Menu* condition is expected be faster. The *T-Menu* condition has a highly significant smaller selection time ($t = 2.86$, $p^{**} = .004$) than the *G-Menu* but not lower than the *K-Menu* condition ($t = 4.80$, $p^{***} = .00003$). The *T-Menu* condition is significantly faster than the *G-Menu* condition if the relative position of the item is under 41. If we take all items (under and above 41), this conclusion cannot be deduced. The *K-Menu* condition stays faster than the *T-Menu*, even with items with a relative position under 41. Therefore, H_{21} is not fully supported. When such an item becomes visible on the main screen, it is not always obvious that the participant will switch to pointing instead of keyword or gesture because this requires some mode switching, which induces some additional work load.

Third Hypothesis: Error Rate. Figure 5 reproduces the error rate aggregated for all participants, then for each participant. We observe that the *K-Menu*

($M = .26$, $SD = .12$) benefits again from the lowest error rate, followed by the *T-Menu* ($M = .36$, $SD = .24$), and the *G-Menu* ($M = 1.04$, $SD = 0.83$). All menus received this a more concentrated interval between the minimum and the maximum values: *K-Menu* ($min = 0$ which means that no errors were produced, $max = .50$), *T-Menu* ($min = 0$, $max = .83$), *G-Menu* ($min = 0$, $max = 3.17$). We can observe that some participants were quite efficient in Fig. 5 when they did not generate any error, which is reflected by a null error rate. Some other participants were on the contrary much more error prone. We have not been able to identify the reasons why some participants remain error-prone vs error-resistant as this behaviour seems to propagate for all menu types, and not for a certain menu in particular. But it is sure that the *K-Menu* has the lowest error rate in all cases. Similarly, we computed three Student's t-tests with two paired samples to determine whether there was any significant difference between the menu types in terms of error rate. There was a highly significant difference in the error rate for *K-Menu* and *T-Menu* conditions ($df = 30$, t = 2.78, $p^{**} = .0047$), for the *K-Menu* and the *G-Menu* conditions ($df = 30$, t = 3.98, $p^{***} = .0002$), and for the *T-Menu* and *G-Menu* conditions ($df = 30$, t = 3.65, $p^{***} = .0005$). Once again, there is a concordance between these results and the questionnaire results. When asked to rank the three menus regarding their easiness (Fig. 4a), 25 participants out of 30 expressed that the *K-menu* is the easiest one to perform the task, which makes sense with the small amount of errors. 19 out of 30 participants thought the *G-Menu* is the least easy menu since it has the highest amount of errors. One potential observation is that participants experienced some trouble to recognize the letters drawn (in particular the letters "g", "q" or "f"), which is mainly due to the gesture recognizer. We gave them an alphabet to show how to write these letters but this did not helped them a lot, they still had difficulties to draw these letters by gesture. The stroke gesture recognition could be trained to learn new end-user defined gestures, which is particularly useful in case of disambiguation. For example, if there is some confusion between the "u" and the "v" letters, which can be detected after issuing correction gesture, the underlying model of the recognition engine implicitly considers that the new alternative may be the correct one and automatically adapts the likelihood accordingly. This feature is considered particularly useful on the long-term, but it was not exploited during the experiment. In conclusion, H_{31} is supported.

5.3   Further Discussion

Although some participants experienced some trouble in producing the gestures, not because they could not, but simply because they did not remember the shape of the stroke, we observed that some women had less trouble in producing the difficult letters because they draw "complexifier" letters like in Fig. 6a, b. Another common error comes with the scrolling: some participants scrolled up and down in the menu and then clicked to quickly on the item, which resulted into the letter "i" being produced Fig. 6c. This could explain why the *T-menu* has significantly more errors than the *K-Menu*. There were also errors made by the condition of the application. Some people saw the letter "G" expressing the
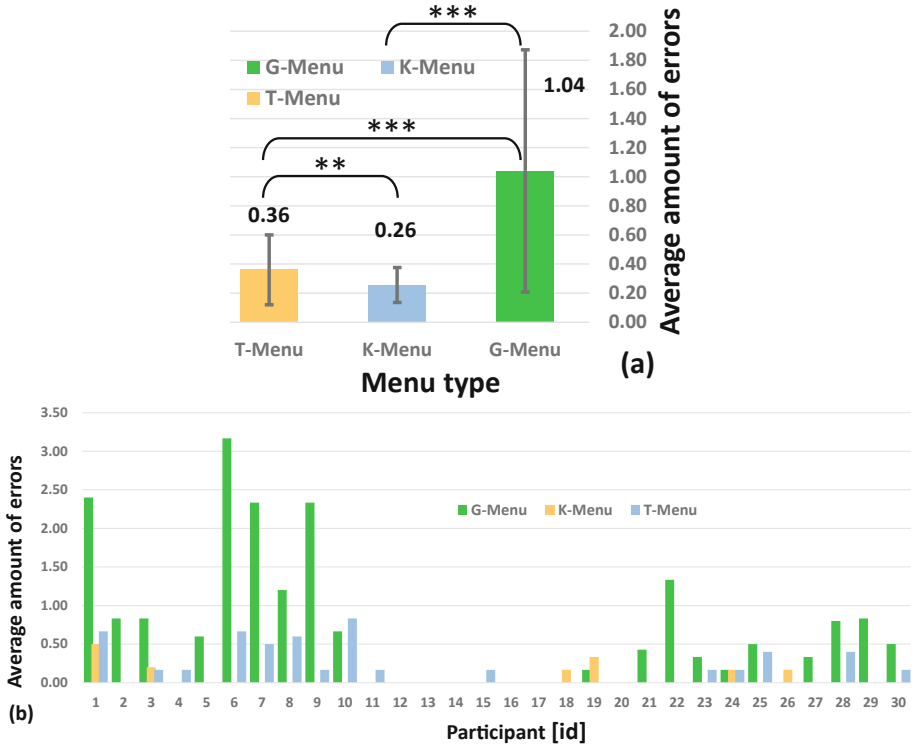
Fig. 5. Average error rate for all participants (a), then per participant (b).

G-Menu on the screen and then began drawing the letter "g" in place of the first letter of the keyword searched. We can call these three types of errors "system" errors. Regarding the five statements reported in Fig. 4, we can see that the general trend is to prefer *K-Menu* to the other menus. People find it more easy, more useful for a task achievement, faster, they felt more productive and it was revealed easier to learn if you work with a *K-Menu* than with the other menus. This goes in the same way that our hypotheses.

So, if the *T-Menu* and *G-Menu* were suggested to be slower and more error-prone than the *K-Menu*, what are its advantages over them? Based on the interviews, we collected some positive user feedback as follows:

– The *G-Menu* always remains at immediate availability and use since the gesture can be issued on any screen, and not just on a particular area as the *K-Menu*. When participants go deeper in the menu structure, coming back to the first or dedicated screen where the keywords can be types requires some swiping time that was not taken into account in the experiment. There was only one screen. This suggest to replicate the experiment with various menu structures.
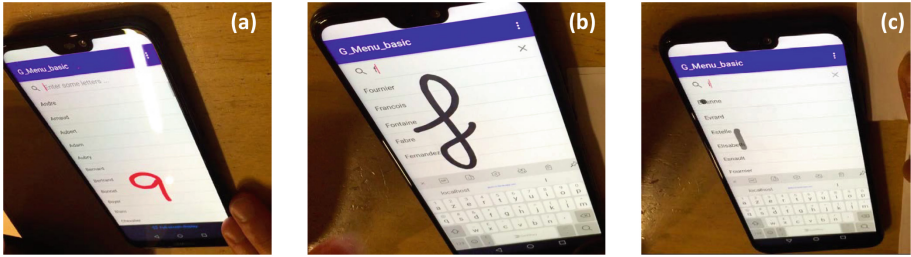
Fig. 6. Cases for gesture recognition.

- The *G-Menu* is always very natural since it is based on stroke gesture recognition of "naturally" produced gestures. The recognition accuracy of recognizers is therefore important in both user-dependent and user-independent scenarios [24].
- The *G-Menu* may be experienced as an enjoyable menu considered as an alternative to the most powerful *K-Menu* when conditions imposed by the context of use are more demanding. Indeed, the *K-Menu* requires reaching a small zone for entering the keyword by tapping, which may cause some trouble for people having some disabilities, like vision or motor impairments. We did not test participant belonging to this population, but we know that gestures need to be adapted to them in terms of articulation. The five statements (Fig. 4) did not cover playfulness or enjoyability, which might be another criteria to consider for the next experiment.

The limitations of the *G-Menu* over the *T-Menu* and *K-Menu* will never be compensated, but could be tackled by offering more significant gestures that are easier to produce, to remember, and to recognize, which is a common problem in gesture recognition.

6 Conclusion and Future Work

This paper presented and compared the G-Menu with respect to the T-Menu and the popular K-Menu against the usual variables of usability: it is slower and more error prone than the others. On the other hand, the experiment did not investigate other variables that go beyond the mere usability and which enters the user experience, such as intuitiveness, playfulness, and immediate usage. These advantages are elsewhere than in the variables controlled in the experiment. The key aspect concerns the stroke recognition: we could also try to find why the G-Menu recognizes some letters with more accuracy than other letters and then exploiting the automatic learning facility of the stroke recognition engine. Another option is to study a composition of these menus: a *GK-menu* which combines gesture and keyword text-entry menus.

References

1. Ahlström, D., Cockburn, A., Gutwin, C., Irani, P.: Why it's quick to be square: modelling new and existing hierarchical menu designs. In: Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2010), pp. 1371–1380. ACM, New York (2010). http://doi.acm.org/10.1145/1753326.1753534

2. Bailly, G., Oulasvirta, A., Kötzing, T., Hoppe, S.: MenuOptimizer: interactive optimization of menu systems. In: Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST 2013), pp. 331–342. ACM, New York (2013). http://doi.acm.org/10.1145/2501988.2502024

3. Bailly, G., Lecolinet, E., Nigay, L.: Visual menu techniques. ACM Comput. Surv. **49**(4), 60:1–60:41 (2017). http://doi.acm.org/10.1145/3002171

4. Bezold, M., Minker, W.: A framework for adapting interactive systems to user behavior. J. Ambient Intell. Smart Environ. **2**(4), 1364–1876 (2010). http://dl.acm.org/citation.cfm?id=2021081.2021084

5. Bouzit, S., Calvary, G., Chêne, D., Vanderdonckt, J.: Polymodal menus: a model-based approach for designing multimodal adaptive menus for small screens. In: Proceedings of ACM Human-Computer Interaction. EICS, vol. 1, pp. 15:1–15:19 (2017). http://doi.acm.org/10.1145/3099585

6. Chêne, D., Petit, É., Zijp-Rouzier, S.: How to achieve design for all: "List", "Focus" and "Multimodality" as minimal requirements. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2016. LNCS, vol. 9738, pp. 117–128. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40244-4_12

7. Findlater, L., McGrenere, J.: A comparison of static, adaptive, and adaptable menus. In: Proceedings of the ACM International Conference on Human Factors in Computing Systems (CHI 2004), pp. 89–96. ACM, New York (2004). http://doi.acm.org/10.1145/985692.985704

8. Fischer, S., Schwan, S.: Adaptively shortened pull down menus: location knowledge and selection efficiency. Behav. Inf. Technol. **27**(5), 439–444 (2008). https://doi.org/10.1080/01449290701497095

9. Fruchard, B., Lecolinet, E., Chapuis, O.: How memorizing positions or directions affects gesture learning? In: Proceedings of the ACM International Conference on Interactive Surfaces and Spaces (ISS 2018), pp. 107–114. ACM, New York (2018). https://doi.org/10.1145/3279778.3279787

10. Fukazawa, Y., Hara, M., Onogi, M., Ueno, H.: Automatic mobile menu customization based on user operation history. In: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI 2009), pp. 50:1–50:4. ACM, New York (2009). http://doi.acm.org/10.1145/1613858.1613921

11. Gajos, K.Z., Chauncey, K.: The influence of personality traits and cognitive load on the use of adaptive user interfaces. In: Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI 2017), pp. 301–306. ACM, New York (2017). http://doi.acm.org/10.1145/3025171.3025192

12. Gobert, C., Todi, K., Bailly, G., Oulasvirta, A.: SAM: a modular framework for self-adapting web menus. In: Proceedings of ACM International Conference on Intelligent User Interfaces (IUI 2019). ACM, New York (2019)

13. Huang, S.-C., Chou, I.-F., Bias, R.G.: Empirical evaluation of a popular cellular phone's menu system: theory meets practice. J. Usability Stud. **1**(2), 91–108 (2006). http://dl.acm.org/citation.cfm?id=2835658.2835662

14. Karlson, A.K., Robertson, G.G., Robbins, D.C., Czerwinski, M.P., Smith, G.R.: FaThumb: a facet-based interface for mobile search. In: Proceedings of ACM Conference on Human Factors in Computing Systems (CHI 2006), pp. 711–720. ACM, New York (2006). http://dx.doi.org/10.1145/1124772.1124878

15. Lee, S.E., Lee, G.: K-menu: a keyword-based dynamic menu interface for small computers. In: Proceedings of Extended Abstracts on Human Factors in Computing Systems (CHI EA 2007), pp. 2543–2548. ACM, New York (2007). https://doi.org/10.1145/1240866.1241038

16. López-Jaquero, V., Vanderdonckt, J., Montero, F., González, P.: Towards an extended model of user interface adaptation: the ISATINE framework. In: Gulliksen, J., Harning, M.B., Palanque, P., van der Veer, G.C., Wesson, J. (eds.) DSV-IS/EHCI/HCSE -2007. LNCS, vol. 4940, pp. 374–392. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-92698-6_23

17. Norrie, L., Murray-Smith, R.: Investigating UI displacements in an adaptive mobile homescreen. Int. J. Mob. Hum. Comput. Interact. **8**(3), 1–17 (2016). http://dx.doi.org/10.4018/IJMHCI.2016070101.oa

18. Peillon, A., Chêne, D., Chaumon, M.-E.B.: HMI and visually-disabled user: the emergence of specific issues. In: Proceedings of 28ème conference francophone sur l'Interaction Homme-Machine (IHM 2016), pp. 259–263. ACM, New York (2016). https://doi.org/10.1145/3004107.3004136

19. Parasuraman, R., Riley, V.: Humans and automation: use, misuse, disuse, abuse. Hum. Factors **39**(2), 230–253 (1997). https://doi.org/10.1518/001872097778543886

20. Petit, E., Chêne, D.: MenuDfA : un composant de navigation gestuelle tactile conçu pour tous. In: Proceedings of 27eme Conf. francophone sur l'Interaction Homme-Machine (IHM 2015). ACM International Series, New York, Article d03 (2015). https://hal.archives-ouvertes.fr/hal-01219909/file/d03-petit.pdf

21. Robbins, D.C., Lee, B., Fernandez, R.: TapGlance: designing a unified smartphone interface. In: Proceedings of the 7th ACM Conference on Designing interactive systems (DIS 2008), pp. 386–394. ACM, New York (2008). http://dx.doi.org/10.1145/1394445.1394487

22. Roy, Q., Malacria, S., Guiard, Y., Lecolinet, E., Eagan, J.: Augmented letters: mnemonic gesture-based shortcuts. In: Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2013), pp. 2325–2328. ACM, New York (2013). https://doi.org/10.1145/2470654.2481321

23. Vanderdonckt, J.: A MDA-compliant environment for developing user interfaces of information systems. In: Pastor, O., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 16–31. Springer, Heidelberg (2005). https://doi.org/10.1007/11431855_2

24. Vanderdonckt, J., Roselli, P., Pérez-Medina, J.L.: !FTL, an articulation-invariant stroke gesture recognizer with controllable position, scale, and rotation invariances. In: Proceedings of the ACM International Conference on Multimodal Interaction (ICMI 2018), pp. 125–134. ACM, New York (2018). http://doi.acm.org/10.1145/3242969.3243032

25. Vanderdonckt, J., Bouzit, S., Calvary, G., Chêne, D.: Exploring a design space of graphical adaptive menus: normal vs small screens. ACM Trans. Intell. Interact. Syst. (TiiS) **9**, 4 (2019)

26. Zhai, S., Kristensson, P.O.: The word-gesture keyboard: reimagining keyboard interaction. Commun. ACM **55**(9), 91–101 (2012). https://doi.org/10.1145/2330667.2330689

27. Zheng, J., Bi, X., Li, K., Li, Y., Zhai, S.: M3 gesture menu: design and experimental analyses of marking menus for touchscreen mobile interaction. In: Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2018). ACM, New York, Paper 249 (2018). https://doi.org/10.1145/3173574.3173823