# Flower Menus: A New Type of Marking Menu with Large Menu Breadth, Within groups and Efficient Expert Mode Memorization

[1,2]Gilles Bailly [2]Eric Lecolinet [1]Laurence Nigay
[1] LIG, University of Grenoble 1, Grenoble, France
[2] TELECOM ParisTech, LTCI CNRS, Paris, France

{Gilles.Bailly, Laurence.Nigay}@imag.fr {Gilles.Bailly, Eric.Lecolinet}@enst.fr

## ABSTRACT

This paper presents Flower menu, a new type of Marking menu that does not only support straight, but also curved gestures for any of the 8 usual orientations. Flower menus make it possible to put many commands at each menu level and thus to create as large a hierarchy as needed for common applications. Indeed our informal analysis of menu breadth in popular applications shows that a quarter of them have more than 16 items. Flower menus can easily contain 20 items and even more (theoretical maximum of 56 items). Flower menus also support *within groups* as well as hierarchical groups. They can thus favor breadth organization (*within groups*) or depth organization (hierarchical groups): as a result, the designers can lay out items in a very flexible way in order to reveal meaningful item groupings. We also investigate the learning performance of the expert mode of Flower menus. A user experiment is presented that compares linear menus (baseline condition), Flower menus and Polygon menus, a variant of Marking menus that supports a breadth of 16 items. Our experiment shows that Flower menus are more efficient than both Polygon and Linear menus for memorizing command activation in expert mode.

## Categories and Subject Descriptors

H5.2. [**User Interfaces**]: Interaction styles. I.3.6. [**Methodology and Techniques**]: Interaction techniques.

## General Terms

Design, Human Factors,

## Keywords

Marking menus, Polygon menus, Flower menus, *within groups*, curved gestures, novice mode, expert mode, learning performance.

## 1. INTRODUCTION

Marking menus [9] are a combination of pop-up radial menus and gesture recognition. Marking menus thus define an interesting alternate solution to Linear menus. However, Marking menus are not yet widely introduced in graphical interfaces. One possible reason is their limit to support an important number of commands: it has been shown that with reasonable accuracy, the limit of hierarchical Marking menus is 64 items (breadth-8, depth-2) [10]. Several variants of Marking menus [1][19][20] have been proposed to partially overcome this limitation: while Multi-Stroke menus [19] focus on the menu depth, Polygon menus [20] increase the menu breadth.

In this paper, we introduce Flower menu, a new type of hierarchical Marking menu, that is designed to contain an important number of commands (>1000). To do so, Flower menus (Figure 1) increase the menu breadth of Marking menus by supporting 7 different curved gestures for each 8 directions. They can then theoretically contain 56 commands at each level. In practice, Flower menus can easily support about twenty commands for a given level (for instance 17 commands in Figure 2-d), which is sufficient for many menu applications: indeed our informal analysis of menu breadth in some popular applications shows that the average number of items per level is 12.4, almost half of the considered applications contained at least 14 items and a quarter of them more than16 items.

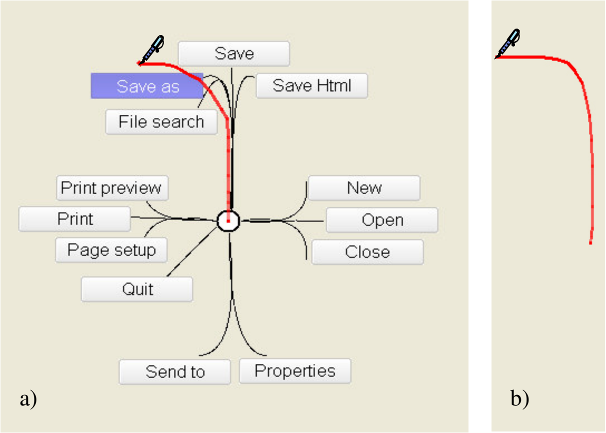
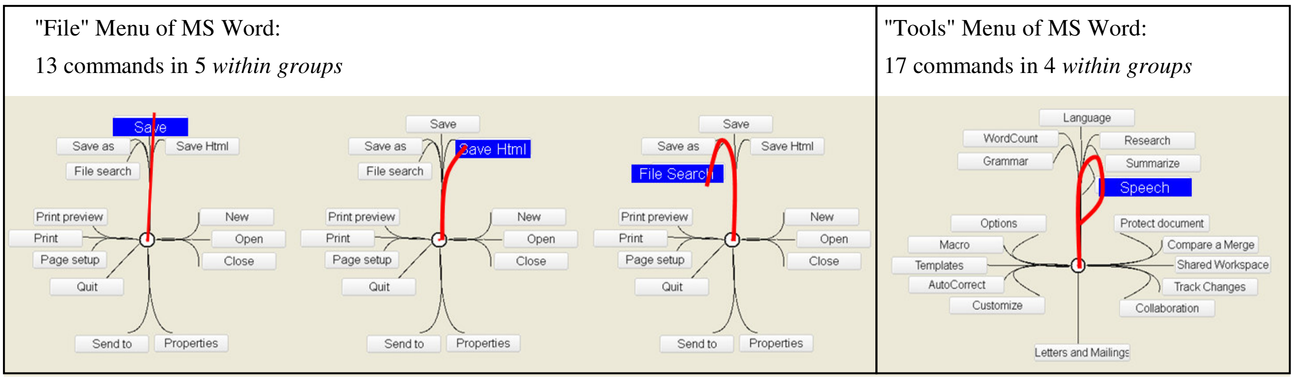


**Figure 1. Flower Menus (a) Novice Mode, (b) Expert Mode.**

In addition to increasing the menu breath, another key feature of Flower menus relies on their ability to support *within groups*. Two types of item groupings are commonly used in menu techniques [13]: *within groups* and hierarchical groups. *Within groups* correspond to item groups at a given level (breadth organization). They are common in Linear menus: such groups are separated by a line (for instance, "New" and "Open" are in the same *within groups* in the "File" menu)". Hierarchical groups



(a) Straight (b) Bent (c) Cusped (d) Pigtail

**Figure 2. Samples of curved gestures in Flower Menus: the Straight (a), Bent (b), Cusped (c) gestures in the "File" menu and the Pigtail gesture (d) in the "Tools" menu of Microsoft Word.**

correspond to item groups across a menu and therefore define the depth organization of a menu. While both *within groups* and hierarchical groups are commonly used in Linear menus, it is surprising to observe that previous studies have never considered *within groups* in Marking menus. Flower menus support *within groups* as well as hierarchical groups. They can thus favor breadth organization (*within groups*) or depth organization (hierarchical groups). In this paper, we focus on 1-level Flower menu: we therefore do not consider hierarchical Flower menus. For a given level, not only Flower menus can support a large number of items, but these items can also be organized in a variety of ways in order to reveal meaningful item groups (i.e., *within groups*). The resulting flexibility in the design of Flower menus is illustrated in Figure 2. The “within group” feature is new in Marking menus and we believe that it makes the menus and items easier to remember and to learn.

The learning of the expert mode is a key point of Marking menus. As users execute the same gesture in novice and expert modes, Marking menus offer a "fluid transition" from the novice to the expert mode: Users learn the expert mode implicitly, just by using the menu repeatedly in novice mode. In contrast, hotkeys (i.e. keyboard accelerators) need to be explicitly learnt by the novice users in Linear menus, and this can slow down the learning process. There are few available experimental studies that compare the learning performance of Linear menus and variants of Marking menus [12]. In this paper, we experimentally investigate the learning performance of expert mode of Flower menus and Linear menus. In our experiment, we also consider Polygon menus since they are one the very few variant of Marking menus that supports more than 8 or 12 items at the same level. Our experimental study shows that Flower menus are more efficient than both Polygon and Linear menus for memorizing command activation in expert mode.

The paper is organized as follows: we first discuss related work. We then present the design of the Flower menus. We finally describe a formal experiment and its results that compare the learning performance of the expert mode of Flower menus with that of Linear and Polygon menus.

## 2. RELATED WORK

As explained in the introduction, Marking menus [9] were introduced by Kurtenbach in an attempt to facilitate the transition from the novice to the expert mode. The novice mode is triggered when the user presses down the pointing device and waits approximately 1/3 of a second. The menu then appears centered around the position of the cursor, allowing item selection by moving in the direction of the desired selection. If the user does not wait and begins dragging immediately, the menu enters into expert mode where the cursor leaves an ink trail. When the user releases the mouse, the gesture recognizer determines the selected item. As novice and expert modes use similar gestures, users should learn the expert mode implicitly, just by performing enough selections in novice mode. Another important feature of Marking menus is that they make possible "eyes free selection" thanks to the scale invariance of interpretation of marks.

The radial layout of Marking menus limits the number of items that can be selected. Performance tends to degrade as menu size increases and 12 items seem to be the maximum to ensure an acceptable error rate [11]. Hierarchical Marking menus have thus been proposed [12] to increase the total possible number of items. Commands can be selected by compound or "zigzag" marks. But this number remains limited: only breadth-8 menus with a depth of at most 2 levels can maintain a reasonable accuracy rate of more than 90%.

Multi-Stroke marking menus [19] define an alternate design that improves the expert mode of hierarchical Marking menus. This technique uses temporal instead of spatial composition: a series of simple inflection-free marks must be drawn instead of a single compound mark. However, while effective in expert mode, this design tends to decrease performance in novice mode. This problem was solved by Wave menus [1], a variant of Multi-Stroke menus that provides optimal performance in both modes.

As explained in [20], the breadth limitation of the different kinds of hierarchical Marking menus may imply awkward groupings of items as well as an increased menu depth. In expert mode, deeper menus require more complex gestures that need more time to be drawn, and are more likely to be badly recognized for traditional hierarchical Marking menus as shown in [12]. In novice mode, the user needs to navigate in a larger number of submenus that may cause disorientation [16]. For these reasons, Zone and Polygon menus [20] have been introduced as a way to extend the menu breadth up to 16 items. These two variants of Marking menus consider both the relative position and orientation of elementary strokes. In the first case, the user first taps to specify the menu origin. This action virtually splits the screen into 4 spatial areas (up/down x left/right relatively to the tap location). Each area corresponds to a different breadth-4 marking menu that the user activates in the usual way. Polygon menus work in a similar way except that the items are the vertices of a N-sided polygon as shown in Figure 2-b. A noticeable consequence is that Polygon menus require “tangential” instead of radial gestures (relatively to the menu origin). Moreover the direction of gestures matters and triggers different commands. Hence, while Zone menus can be seen as a kind of hierarchical radial menu, Polygon menus indeed follow quite a different design. Both techniques were reported to have good performance for selecting items, although slightly slower than Multi-Stroke marking menus. But this was globally compensated, considering the fact that regular breadth-8 Multi-Stroke marking menus would require an increased depth for providing the same number of items.

A common point of all these studies is that they only evaluated the performance for selecting items in expert mode. While Marking menu techniques and their variants seem likely to favor the transition from the novice to the expert mode, we found only one study that attempted to verify this hypothesis experimentally [12]. The experiment focuses on the behavior of two users of an extended real application over a long period of time (i.e., hundreds of hours). In this setting, results demonstrate the effectiveness of Marking menus over Linear menus and show the gradual transition from the novice mode to the expert mode. Nevertheless the study was performed with two users only. Moreover alternate design of Marking menus such as Polygon menus are different enough from the original Marking menus to lead to significantly different results. This point motivated our experimental study on learning performance of the expert mode. Before describing the conducted experiment and its results, we first present the Flower menus, that we have considered in our experiment.

## 3. FLOWER MENU DESIGN

Flower menus[1] extends Marking menus by making it possible to draw straight or curved gestures. As with standard Marking menus, the user must press the mouse, perform a radial gesture and release the mouse. The user always starts a gesture from the same point (i.e. the menu center in novice mode) and no tap is needed to specify the menu origin as is the case for Polygon menus. This property is important as users reported that they prefer gestures “starting from the center” in Flower menus rather than “having to perform two operations” in Polygon menus.

Althought used in a different context, curved gestures have been proposed in menuing systems [6] or for entering text [8]: closed loops in the first case and bent gestures on the on-axis in the second case.

In addition to orientation, curvature provides a complementary way to encode input data. Flower menus make the most of possibilities to increase the number of available commands while making them easy to perform. In order to fullfill this criterion, we retained 4 different degrees of curvature. Considering the rotating direction (clockwise, counterclockwise), Flower menus provide 7 gestures for each 8 directions (Figure 3 shows them for the North orientation):

- S: a *straight* gesture, as in regular Marking menus,
- B-,B+: *bent* gestures, that can either be curved in the clockwise or counterclockwise direction, as in the “hybrid design”,
- C-,C+: *cusped* gestures, that can also be curved in both rotational directions,
- P-,P+: *pigtail* gestures, considered in both rotational directions.

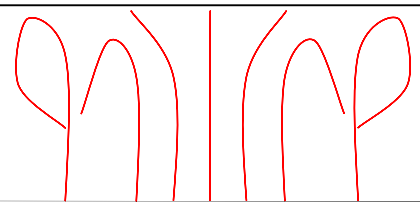
| Gesture |  | | | | | | |
|---|---|---|---|---|---|---|---|
| Name | P- | C- | B- | S | B+ | C+ | P+ |
| Execution time (ms) | 937 | 817 | 708 | 498 | 704 | 813 | 931 |

**Figure 3. The 7 gestures of Flower menus for the Northern orientation and their average execution times.**

A Flower menu can thus contain a theoretical maximum of 7 * 8 = 56 items for each level (practically, Flower menus can contain approximately 20 items). While most menus will obviously not contain so many items, this feature is most useful for creating *within groups*, enabling the designer to choose amongst a large variety of spatial organisations. For example, Figure 2 shows how to organize the 5 *within groups* of the Microsoft "File" menu in a Flower menu. Moreover since Flower menus support both *within groups* and hierachical groups, the designer has even more possibilities for spatial arrangments, balancing breadth organization (*within groups*) and depth organization (hierarchical groups). Meaningful groups make easier the learning and memorization of commands. Indeed commands of a particular group are semantically related and such a semantic relationship is stored in the human declarative memory. This is possible because of the spatial arrangement of commands in a Flower menu group. Commands are spatially close in a “petal”: such proximity and closure are two Gestalt principles.

About memorization, it is also worth noticing that Flower menus are based on a highly symetrical design. They use 4 different types of gestures (*Straight, Bent, Cusp, Pigtail*), that can be drawn along 8 different orientations and curved in 2 different ways (except for *Straight* lines, where the curvature is null). These 4 gesture types can also be seen as a variation of the same drawing: a line that is more and more curved. As a consequence, users can consider and remember the 56 theoretical possible positions of a Flower menu as a combination of 3 variables having at most 8 possible values (i.e. 8 orientations x 4 types x 2 rotating directions). This point may be an important factor for memorization, as explained in the discussion section.

Finally, hierarchical Flower menus work in the same way as Multi-Stroke Menus [19]. Both menus support a series of overlapping marks (Figure 4) rather than the kind of single zigzag marks used in original Hierarchical Marking menus (HMM). Mutli-Stroke menus have been shown to be as fast and less error prone than HMMs [19], especially for large menu systems as HMMs tend to produce many errors for diagonal gestures.

---

[1] A video can be found at: www.gillesbailly.fr

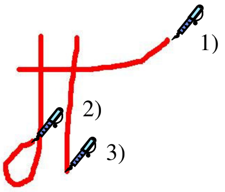


**Figure 4. A selection with a 3-level Flower menu (that generalizes Hierarchical Multi-Stroke Menu) in expert mode. 1) bent, 2) pigtail and 3) straight marks.**

# 4. PILOT STUDY

We conducted a pilot experiment to study how users perform Flower gestures. The experiment is fully described in [2]. The expected outcome of our experiment was: a) to obtain experimental data in order to develop an effective gestures recognizer; b) to verify that users could draw all these gestures precisely enough c) to find the most efficient gestures for the design of a flower menu by identifying where frequently used items should be prefrentially placed in the menu. Besides, the gesture database that was produced during this experiment was then used to train and to test the recognition algorithm. The 14 right-handed participants were asked to draw as quickly and accurately as possible 56 gestures (8 directions * 7 gestures). To illustrate the conducted experiment, Figure 5 shows all the performed "Bent" gestures drawn by all the participants for the counterclockwise direction.

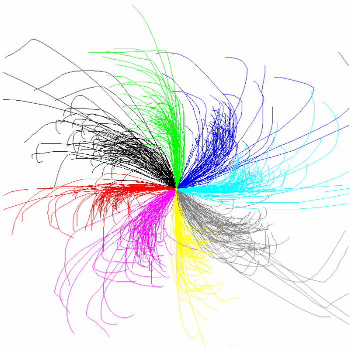


**Figure 5. Bent gestures for the counterclockwise**

As expected by the two-thirds power law [18], our results show that drawing time grows with curvature: straight lines (498 ms) are faster than bent (704 ms), cusp (813 ms) and pigtail (929 ms). The most frequent commands should thus preferentially be placed on straight and bent lines. These results are coherent with those of [3][20] for comparable gestures. However, it is interesting to notice that the times obtained in all these studies is higher than in [3] whose experiment favors speed because it is based on very repetitive movements. The actual speed obtained by very trained users of Flower menus may thus be shorter than in our results.

We did not consider inflexions (corner gestures) in our experiment for two reasons: this would have made it too long and inflexions have been shown to be slower than bent gestures [3]. For this reason, the 16 first commands of Flower menus (straight and bent gestures) should be faster than the 16 commands in a 2-level Marking menu as these ones require inflexion gestures (besides, 2-level Marking menus do not provide equivalent capabilities as all items can not be seen at the same time).

We also observed that the angular variability is higher on the off-axis orientation (diagonals). As a consequence, large groups should be preferentially put on the on-axis orientation of the Flower menus. However, this effect can be largely compensated by an effective recognizer taking into account the actual size and position of the angular sectors of circular menus.

So instead of considering a naïve algorithm that would not take precisely into account how users draw marks and would thus misinterpret some correct gestures, we developed a specific recognizer (based on K-nearest neighbors) which is both fast and effective. We used the samples drawn by one half of the participants for training and the other half for testing. We also removed gestures that were erroneously drawn from the database (about 2% of all gestures).

The recognizer is fast enough to provide immediate feedback. The overall recognition rate is 99% for the first 24 commands (straight + bent gestures); 96.5% for the first 40 commands (cusped gestures added) and 93% for all the commands. However, for the case of real applications, pigtail gestures corresponding to the case of *within groups* of 6-7 items will not be very frequent. The real recognition rate will thus certainly be superior to 96.5%. The tuning of the gesture recognizer was a prerequisite for the experiment presented in the following section. The samples of the testing set were merged with those of the learning set to obtain a larger learning database.

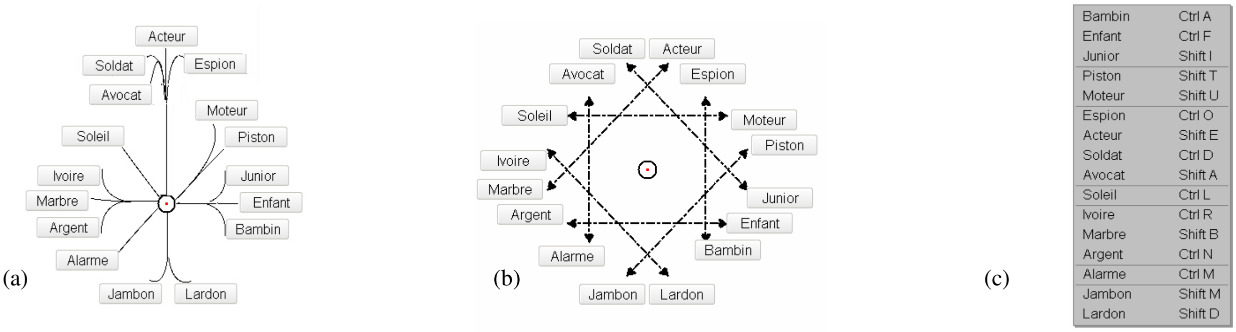
# 5. EXPERIMENT

The goal of this experiment was to compare the learning performance of the expert mode of Flower menus with Linear menus (baseline) and Polygon menus. In this experiment, we focus on the comparison of three significantly different menu techniques that can contain at least 16 commands at the first level as it is often the case in existing applications. We did not consider traditional 2-level Marking menus nor 2-level Multi-Stroke menus in this experiment for three reasons: a) these techniques do not allow to display many items at a single level; b) this would have introduced another variable (the menu depth) in the experiment; and c) this would have made our experiment too long. However, comparing the performance of 1- and 2-level Flower menus would be an interesting track for future work as Flower menus both generalize Multi-Stroke menus and provide more items.

## 5.1 Menu Configuration

We designed a "canonical" menu configuration (Figure 6) that is intended to be representative of those seen in real applications. For this purpose, we performed an informal analysis of the content of pull down menus (more precisely, the first-level pull down menu in menu bar) in popular applications for MS-Windows (Table 1).

According to this informal analysis, the average breadth is 12.4 items, 46% of the menus contain at least 14 items and 23% of them more than 16 items. As explained in [20], these results confirm the need for increasing marking menu breadth. They led us to perform our experiments with breadth-16 menus, breadth-16 being also the maximum size for Polygon menus. Furthermore, in our informal analysis, we studied the frequency of *within groups* depending on their size (Table 1). All menus have *within groups*, 58% of these groups contain 1 or 2 items and 92% of them up to less than 4 items. This led us to adopt a menu configuration with similar statistics (Figure 6): Two 1-item groups; two 2-item groups; two 3-item groups; and one 4-item group. In this design, the percentage of 1-2-3- and 4 item groups is close to the results of our analysis. Besides, this configuration (shown in Figure 6) is



(a) (b) (c)

**Figure 6. (a) Flower, (b) Polygon, and (c) Linear menu configurations used in the experiments.**

almost identical to the File menu in word 2003 for Windows. According to the results of our pilot study (angular variability being larger on the diagonals), we placed the largest groups on the on-axis orientations of the Flower menu (Figure 6-a). We also placed same sized groups in different areas of the menu to avoid layout singularities. Groups were placed in the same order in the Polygon menu (Figure 6-b) and the Linear menu (Figure 6-c), starting from the NW position of the Flower menu and by following the counterclockwise direction.

| Application | nb items | menu ≥ 3 groups (%) | groups ≤ 2 items (%) | groups ≤ 4 items (%) | groups ≤ 7 items (%) |
|---|---|---|---|---|---|
| excel 03 | 13.3 | 89 | 37 | 92 | 100 |
| adobe reader 7.0 | 10.6 | 86 | 68 | 94 | 100 |
| word 03 | 14.2 | 89 | 45 | 85 | 100 |
| firefox 2.0 | 8.9 | 100 | 72 | 92 | 100 |
| thunderbird 0.9 | 9.4 | 100 | 61 | 94 | 100 |
| photoshop 7.0 | 18 | 100 | 66 | 94 | 99 |
| mean | 12.4 | 94 | 58.17 | 91.83 | 99.83 |

**Table 1: Informal analysis of pull-down menus in some applications for MS-Windows.**

## 5.2 Items and groups

The design of Flower menu makes groupings implicitly visible. We slightly changed the positions of items in the Polygon menu in order to reveal groups (so that items belonging to a same within group would be slightly closer). We used regular separators in the Linear menu. Each group contains items corresponding to a given category such as colors, animals, music, transportation means, etc. These categories and the item names were carefully chosen to avoid possible ambiguities (so that an item could not belong to multiple categories). All item names are 6 letters long and do not contain rare French letters such as Q, Z, Y, W, H.

## 5.3 Linear menu hotkeys

Keyboard hotkeys were assigned to items in a way that attempted to be as realistic as possible while avoiding undesirable singularities that could bias the results. For this purpose, we realized an informal analysis of hotkeys in Microsoft Word and FireFox. This study showed that there is a great variety of hotkeys and that the hotkey letters is not always part of the item name. For example, Ctrl+D activates the "Font" command in Word and Ctrl+F2 the "Print Preview" commands in Firefox. However, we decided to make the task simpler for our participants because many of them complained in a preliminary experiment where hotkeys were not always contained in the corresponding item name. Hotkey letters are thus part of the name in our experiment with the exception of the first and last letter to avoid making certain items easier to remember.

We also discarded C, V, X, and Z because some users developed specific strategies to remember the mapping between items and hotkeys with these specific letters. This effect, probably caused by the high familiarity of users with these keys, would have introduced undesirable variability. We only used Ctrl and Shift as modifier key although other modifiers and combinations of them are common in real applications (especially on the Macintosh where commands such as Shift+CMD+DEL, Alt+CMD+M, Alt+Shift+CMD+C,… are widely available). A consequence of this design is that we do not only use keys located on the left side of the keyboard (as done in some previous studies [5]) because they are not enough of them to match 16 items without breaking the previous constraints.

As a conclusion, Linear menus were tested in rather favorable case in our experiment. A real life application that would attempt to associate as many possible hotkeys to commands: a) could not use the first letter or even simply a letter of the word for most commands because of name collisions; b) could not use well known hotkeys because they are already used for standard operations; and c) would thus have to use all possible letters, symbols and function keys and a variety of modifier key combinations.

## 5.4 Stimulus

The stimulus was the name of the item that the user had to select. We used a textual stimulus, rather than an iconic one, in order to avoid possible confusion since the items are grouped according to semantic relationships.

We did not use a Zipfian distribution [5] but a uniform target frequency. This is because the memorization of items may depend on ordering (for Linear menus), on orientation (for the two marking menus), and on type (for Flower menus). A Zipfian distribution would thus make results dependent on where the most frequent items are placed in the 3 types of menus. A uniform distribution avoids this problem and makes results comparable with the 3 menu techniques.

## 5.5 Hypothesis

*H1: Markings menus (i.e. Flower and Polygon) favor expert mode memorization* because the same actions are performed in novice and expert mode.

*H2: Expert mode memorization is better with Flower than Polygon menus*. Flower menus with explicit *within groups* make the mapping between gestures and orientations very straightforward, a feature that may help memorization.

*H3: Linear menus are faster than Flower menus that are faster than Polygon menus in expert mode.* Linear menus should outperform marking menus on this criterion because hotkey activation should require less time than drawing a gesture [17]. The average performance of Flower menu gestures should be higher than Polygon gestures for a well-balanced 16 item menu (that is to say a menu where items are not arbitrarily put on the slowest locations).

## 5.6 Procedure

In this controlled experiment, we intend to evaluate the learning of expert mode, by comparing how many items the users are able to select in expert mode. More precisely, the purpose of our study was to evaluate the *intentional* learning of the expert mode as opposed to *implicit* learning since users were explicitly asked to learn the expert mode. Nevertheless a design that makes it easier to remember the expert mode also favors its implicit learning.

We chose to evaluate *intentional* rather than *implicit* learning because this latter condition is, by nature, imprecisely defined. It is in fact quite difficult to evaluate implicit learning in a controlled experiment because these conditions are likely to influence how the users learn the expert mode. A longitudinal user experiment within the context of a real-world application as described in [12] would be necessary for studying implicit learning. Indeed, based on our previous observations and those from previous studies [5], users adopt different strategies for learning the expert mode, especially in the context of a real task.

Our experiment roughly follows the design of the *memory recall task* in [4] and comprises three different phases.

*Familiarization*. The familiarization process consists of explaining how the tested technique works in novice and expert mode and allows for user practice in order to be sure s/he knows how to operate. This phase took about 2 mn.

*Training*. Participants where instructed that the goal was to learn how to select as many items as possible in expert mode. They were told not to "rush" in selecting items because time was unimportant in this phase and excessive speed would degrade their performance in the testing phase. They were then asked to select items during 5 mn, first in novice mode to learn them, then in expert node when they felt able to do so. The same item was presented again in case of a wrong selection. Otherwise, the stimulus was chosen according to a random distribution (except that an item could only appear once in a 16 stimuli sequence).

*Testing*. Participants were asked to correctly select items in expert mode as fast as possible, the novice mode being disabled. The stimulus was the same and the 16 possible items were presented in random order. This phase was repeated twice in order to get more experimental data in order to evaluate the time performance. During this phase, no feedback was provided to indicate if the selection was correct. Nevertheless we gave participants a second chance to learn the menu: between two blocks, the menu was displayed again for 15 seconds.

## 5.7 Design

The ordering of the three techniques was counterbalanced across subjects using a Latin square design. Three equivalent sets of item names were used to avoid transfer effects between the first, second and third tested technique. As these three sets were chosen to be semantically equivalent, this should not have a noticeable effect. However, we also counterbalanced sets with techniques and orderings so that all the techniques would be tested with the same conditions. Each participant performed the experiment in one session which was about 40 mn long. In summary, the design was as follows:

18 participants **x**
3 menu techniques **x**
16 gestures **x**
2 blocks
= 1728 selections.

## 5.8 Participants and Apparatus

18 participants (3 female) ranging in age from 22 to 35 years (mean 26) were recruited from within the university community and received a handful of candies for their participation. They were all right-handed and familiar with computers. The experiment was conducted on a Dell Latitude D800. The experimental software was implemented in C++/Qt. Participants used a 3 button Logitech mouse. A mouse was used, rather than a tablet's stylus, for two reasons: the mouse is still by far the most commonly used input device and previous studies showed that equivalent or better results are obtained by using a stylus [10]. By performing our experiment in the "worst case" we wanted to demonstrate the robustness of Flower menus as well as their efficient usage with common input devices.

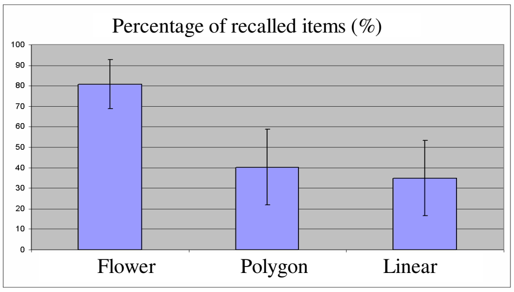


**Figure 7. Percentage of recalled items for the 3 menu techniques.**

## 5.9 Results

As expected, a 4-way analysis of variance shows that the item sets have no significant effect on memorization or selection time.

### *5.9.1 Expert mode memorization*

Analysis of variance reveals a significant main effect for techniques on the **number of recalled items** ($F_{2,34} = 70.34$, $p < 0.0001$). A post hoc Tukey test with 5% alpha level shows (Figure 7) that Flower menus, with 81% of recalled items (12.9/16), are better than Polygon menus (40%; 6.4/16) which are better than Linear menus (35%; 5.5/16). Hypotheses H1 and H2 are thus verified, but we expected a smaller difference between Polygon and Flower menus (as they are both marking menus) and a larger difference between Polygon and Linear menus.

Analysis of variance also shows an effect for testing order ($F_{2,34} = 5.69$, $p < 0.01$). The number of recalled items is globally higher for techniques tested in second rank (9.5/16) than in first (7.3) and third (8.0) ranks, but there is no [technique x order] interaction.

### *5.9.2 Activation performance*

The time required to activate commands comprises two components: the reaction time (interval between the appearance of the stimulus and the mouse down) and the execution time (drawing time). ANOVA indicates a significant effect for technique on **execution time** ($F_{2,34} = 21.58$, $p < 0.0001$). A post hoc Tukey test with 5% alpha level shows that Linear Menus (0.6 seconds) are faster than Flower Menus (0.8 s) that are faster than Polygon menus (1.7 s).

While the results for the execution time correspond to our hypothesis (H3), this is not the case for the reaction time. ANOVA shows that the **reaction time** ($F_{2,34} = 9.07$, $p<0.001$) is significantly longer for Linear Menus (2.9 s) than for Polygon Menus (2.1 s), and is longer for Polygon Menus than Flower Menus (1.6 s). These results suggest that the mapping between commands and hotkeys were less well learned than between commands and gestures (Flower gestures being especially efficient).

ANOVA reveals a significant effect for technique on total time ($F_{2,34} = 7.34$, $p < 0.01$) that indicates that Flower Menus (2.4 s) are faster than Linear Menus (3.5 s) and Polygon menus (3.8 s). Hypothesis H3 is thus not completely verified as Linear menus are slower than Flower menus.

Finally, ANOVA also indicates a significant effect for block on reaction time ($F_{1,17} = 3.76$, $p< 0.001$) and total time ($F_{1,17} = 14.14$, $p < 0.001$). Block 2 is faster than block 1 both for reaction (2.0 vs. 2.4 s) and total time (3.0 vs. 3.4 s).

### *5.9.3 Subjective preference*

In a post-experiment questionnaire, participants ranked the three menu techniques as follows: Flower, Linear and Polygon (in preference decreasing order). 17/18 subjects chose Flower Menus as their favorite technique. We also asked their opinions about the following criteria: familiarization, simplicity, learning, speed, accuracy and fun according to a 5 pt Likert scale. Flower obtained the highest value for all criteria except accuracy. ANOVA followed by a pairwise comparison reveals that: Flower and Linear menus are significantly faster than Polygon menus for familiarization (F:4.6; P:3; L:3.9), speed (F:4.3; P:3.1; L:3.9) and simplicity (F:4.5; P:3.1; L:4.4). Logically, Linear menus (4.8) are significantly more accurate than Polygon (3.9) and Flower (3.7) menus. The “recall” criterion was significantly higher for Flower menus (4.4) than for Polygon (2.2) and Linear menus (2.3). Finally, Flower menus (4.7) are more fun than Polygon menus (3.3) that are considered more "fun" than Linear menus (2.1).

Finally, most users said they preferred gestures “starting from the center” with Flower menus rather than “having to perform two operations” (hence referring to the initial tap of Polygon menus). Most of them found it easy to learn and perform Flower gestures. One user summarized up a general feeling as follows: "I make the general orientation, then I adjust". Some users found it difficult to “learn two things” in Polygon menus, “the position of the item and the gesture". Others noticed that they “knew the position of the command but could not recall the gesture” in Polygon menus.

## 6. Discussion

*Activation performance*. Our results on Total time indicate that Flower menus (2.4 s) are faster than Linear menus (3.5 s) and Polygon menus (3.8 s). The difference between Flower and Linear menus performance is caused by a much longer reaction time in the case of Linear menus. This point suggests that hotkeys were well less learned than gestures in our experiment. However, it is interesting to remark that the reaction time is overestimated and the execution time is underestimated for Linear menus. This is because, the amount of time needed for moving the hands to press the hotkeys should theoretically be counted in the execution time, but this was not technically feasible in our experiment.

Another important remark is that our experiment was not conceived to evaluate activation performance but user capability to learn the expert mode of these menus. The activation times we obtained give interesting indications for comparing the relative performance of these three kinds of menus but they should not be interpreted as the actual times that would be obtained for trained users in expert mode. Both reaction and execution time would be shorter. For instance, execution times are about 20% faster in our pilot study where the task was closer to expert usage.

*Memorization performance.* Our study clearly shows that Flower menus are more effective than both Polygon and Linear menus for memorizing command activation in expert mode: Flower menus are twice more efficient than Polygon menus (12.9 vs. 6.4 items) which are themselves better than Linear menus (5.6 items). As explained in section 5.6, it is important to recall that our experiment evaluates the intentional learning of the expert mode as opposed to implicit learning since users were explicitly asked to learn the expert mode. However, the fact that Flower menus make it possible to remember the expert mode in a short amount of time suggests that users will be very likely to learn it implicitly. This contrasts with Linear menus where many users never learn the expert mode (or only very few hotkeys) because it differs from the novice mode.

While these results validate our hypotheses for learning efficiency (H1, H2) they do not exactly correspond to what we initially expected. In fact, as Flower and Polygon are both marking menus, we expected a smaller difference in performance between them, and a larger difference between Polygon and Linear menus. The following paragraphs provide some possible explanations.

First, the better memorization performance of Flower menus as compared to Polygon menus may result from a simpler mapping between gestures and orientations. As for the original marking menu design, Flower gestures are radial and thus start from the menu center so that users only have to recall the orientation of gesture endings. In contrast, Polygon menus use “tangential” gestures that involve a spatial mapping that is more complex (noticeably, Polygon menus also require the users to remember from which direction the gesture must start).

This point suggests that the directness of the mapping between gestures and spatial orientation is a major factor for the efficiency of marking menus. The argument that is usually put forward to explain why marking menus are better than Linear menus is that users learn the expert mode implicitly by repeating the same gestures in novice mode. This effect may be overestimated, and the main reason why people can easily learn the expert mode of radial marking menus may be just that their expert mode is just very easy to learn. This is in fact what our results suggest. Both Flower and Polygon menus are based on this idea of learning by repeating gestures, but only the radial design (i.e. Flower menus), that provides an easy-to-learn straightforward spatial mapping, gave much better results than Linear menus.

However, Flower menu do not only require users to recall orientations but also the curvature and the rotational direction of gestures. Our experiment showed that users had no difficulty in remembering this combination of 3 different attributes (at least for activating a set of 16 different commands). This result may be explained by the item grouping feature of Flower menus and the “Magical number seven” of the theory of Miller [14] that states that: a) there are approximately only 7 different values that can be distinguished by users for performing a one-dimensional judgment, and; b) this number can be greatly increased by considering a set of independent variable attributes. In other words: “we can make relatively crude judgments of several things simultaneously” [14]. The design of Flower menus fits very well with this principle as it makes use of 3 different attributes having few possible different values (8 orientations x 4 curvatures x 2 rotating directions).

*Combining hotkeys and marks*. Finally it is important to notice that hotkeys and marks are not incompatible. Although our experiment compares marks with hotkeys, it is possible to combine these two functional expert modes: hotkeys and marks will then be redundant, defining two different ways to activate a command. By doing so, introducing Flower menus in an application will not conflict with previous habits. Moreover as for hotkeys across different applications, some flower gestures should remain the same in different applications, resulting in a common gesture vocabulary with straight gestures for frequent commands.

# 7. CONCLUSION

We have presented Flower menus a new type of hierarchical Marking menus that does not only support straight, but also curved gestures for any of the 8 usual orientations. Flower menus make it possible to put many commands at each menu level (they can easily support about 20 commands and even more) and thus to create as large a hierarchy as needed for common applications. Flower menus also support *within groups* as well as hierarchical groups. They can thus favor breadth organization (within groups) or depth organization (hierarchical groups): as a result, designers can lay out items in a very flexible way in order to reveal meaningful item groupings. Flower menus also conserve the advantages of classical Marking menus like "scale independence" and "eyes free selection".

Focusing on the learning performance of the expert mode, we have presented a comparative study of Flower, Linear and Polygon menus. The conducted experiment showed that Polygon and Flower menus offer better performance for learning the expert mode as compared to Linear menus. Moreover the Flower menus resulted in better performance for activation and more importantly for learning the expert mode than Polygon menus. Flower menus are thus a very efficient technique for large breadth menus. They now make possible the use of Marking menus in a wide range of conditions and are well suited for applications that require menus with many items and *within groups*.

There are several directions for future work. In addition to the study of implicit learning of the expert mode in a longitudinal experiment, we plan to compare 1- and 2-level Flower menus to study the design tradeoff between breadth organization (*within groups*) and depth organization (hierarchical groups) and its impact on learning performance.

# 8. ACKNOWLEDGMENTS

Many thanks to Y. Guiard, M. Tahir, A. Roudaut, and S. Malacria for numerous discussions about the Flower menus, and G. Serghiou for his help with the English

# 9. REFERENCES

[1] Bailly, B., Lecolinet, E., Nigay, L. (2007). Wave Menus: Improving the Novice Mode of Hierarchical Marking Menus, INTERACT'07. Springer. P. 475-488.

[2] Bailly, G., Lecolinet, E., Nigay, L. (2007). Analysis of curved gestures. Technical Report GET/ENST.

[3] Cao, X. and Zhai, S. (2007). Modeling human performance of pen stroke gestures. In ACM CHI '07. p. 1495-1504.

[4] Cockburn, A, Kristensson, P, Alexander, J and Zhai, S (2007). Hard lessons: effort-inducing interfaces benefit spatial learning. ACM CHI'07. p. 1571-1580.

[5] Grossman T., Dragicevic, P., Balakrishnan, R. (2007). Strategies for accelerating on-line learning of hotkeys. ACM CHI'07. p. 1591- 1600.

[6] Guimbretière, F., Winograd, T. (2000). FlowMenus: combining command, text and data entry. ACM UIST'00. p. 213-16.

[7] Helson, H. (1933). The fundamental propositions of gestalt psychology. Psychology Review 40, p. 13-31.

[8] Isokoski, P. Käki, M. (2002). Comparison of two touchpad-based methods for numeric entry. ACM CHI'02, pp. 25-32.

[9] Kurtenbach, G., Buxton, W. (1991). Issues in Combining Marking and Direct Manipulation Techniques, ACM UIST'91. pp. 137-144.

[10] Kurtenbach G., Buxton, W. (1993). The limits of expert performance using hierarchical marking menus. ACM CHI'93. pp. 35-42.

[11] Kurtenbach, G., Sellen, A., Buxton, W. (1993). An empirical evaluation of some articulatory and cognitive aspects of marking menus. Journal of Human Computer Interaction, 8(1), p. 1-23.

[12] Kurtenbach, G., Buxton, W. (1994). User learning and performance with marking menus. ACM CHI'94. p. 258-64.

[13] Lee, E.S., Raymond, D. R. (1993). Menu-Driven Systems. Encyclopedia of Microcomputers, Vol. 11, p. 101-127.

[14] Miller G.A., (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. The Psychological Review, 63, p. 81-97

[15] Moyle, M., Cockburn, A. (2002). Analysing Mouse and Pen Flick Gestures. CHI'02, p. 19-24.

[16] Norman, K. (1991). The Psychology of Menu selection: Designing Cognitive Control at the Human/Computer Interface. Ablex Publishing Corporation.

[17] Odell, D. L., Davis, R. C., Smith, A., and Wright, P. K. (2004). Toolglasses, marking menus, and hotkeys: a comparison of one and two-handed command selection techniques. GI'04, p. 17-24.

[18] Viviani, P., Terzuolo, C. (1982). Trajectory determines movement dynamics. in Neuroscience, 7(2). 431-437.

[19] Zhao, S., Balakrishnan, R. (2004). Simple vs. compound mark hierarchical marking menus. ACM UIST'04. pp. 33-44.

[20] Zhao, S., Agrawala, M., Hinckley, K. (2006). Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. ACM CHI'06. p. 1077-1087.