

---

**Kurtenbach, G., Sellen, A. & Buxton, W. (1993).** An empirical evaluation of some articulatory and cognitive aspects of "marking menus." *Human Computer Interaction*, 8(1), 1-23.

---

# **An empirical evaluation of some articulatory and cognitive aspects of "marking menus.**

**Gordon P. Kurtenbach, Abigail J. Sellen, and**

**William A. S. Buxton**

*Computer Systems Research Institute,*

*University of Toronto*[\[\\*\]](#)

## **ABSTRACT**

We describe "marking menus", an extension of "pie menus" which makes . Pie menus are circular menus subdivided into sectors, each of which might correspond to a different command. Marking menus are pie menus in which the path of the cursor leaves an ink trail. Thus, selecting a sector from a marking menu creates a visual mark similar to a pen stroke on paper. Marking menus are also unique in that they ease the transition from novice to expert user. Novices can "pop-up" a menu and make a selection, whereas experts can simply make the corresponding mark without waiting for the menu to appear.

This paper describes an experiment designed to explore both articulatory and cognitive aspects of pie and marking menus. "Articulatory aspects" refers to how well subjects could execute the physical actions necessary to select from pie menus, given three different kinds of input devices (mouse, trackball, and stylus), and as the number of items in the menu increases. Articulatory aspects were investigated by presenting one group of subjects with the task of selecting from fully visible or "exposed" menus. To investigate the cognitive aspects, two other groups of subjects used invisible or "hidden" pie menus: one group with an ink trail, and one without. In order for marking menus to work effectively, users must be able to mentally represent and associate Selection from hidden menu was designed to reveal Both number of slices

per menu and input device were systematically varied. We discuss the findings with respect to menu size, input device, analysis of markings used, and learning.

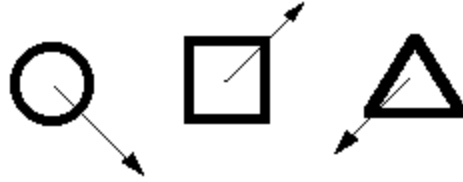
## 1. INTRODUCTION

Computer interfaces which use the stylus as the primary input device have recently begun to receive a great deal of attention (e.g., Leitch, 1990; Normile & Johnson, 1990; Rebello, 1990). Part of the reason is the promise of small portable "book-like" computers and large "whiteboard" sized displays. Because both are particularly well-suited to stylus input, it is important to explore effective ways of supporting stylus-driven interaction in the interface.

Most of us have a lifetime of experience marking paper with pens and pencils. Using a stylus, therefore, can allow us to exploit these everyday skills. Most other input devices constrain the extent to which we can draw on these skills, as trying to sign one's name with a trackball effectively demonstrates. The everyday skills of making markings need not be limited to the domain of free-hand writing and drawing. Computer applications have been developed, or are being developed, that use markings to perform tasks such as text editing, constructing musical scores, mathematical equation formatting, and graphics layout (Carr, 1991; Hardock, 1991; Rubine, 1991; Welbourn & Whitrow, 1988; Wolf, Rhyne, & Ellozy, 1989).

Markings can also provide a quick way to invoke commands. For example, in GEdit (Kurtenbach & Buxton, 1991), a user can make simple shorthand strokes to simultaneously create and position graphical objects. Since the application demands frequent creation of these objects, doing so with minimum effort enhances the interaction. In GEdit, these simple marks can be articulated very quickly (see Figure 1).

A major problem using markings in this way, however, is that the marks are non-mnemonic -- learning and remembering which commands correspond to which marks is difficult. One solution we will describe is to supply the user with a simple way of finding out what marks can be used and what they do. When the system itself supplies this information through the mechanism used to invoke commands, we refer to this as being *self-revealing*. Menus and buttons, for example, are self-revealing: the set of available commands is readily visible as a by-product of the way commands are invoked.



**Figure 1.** Marks for adding objects in GEdit. Three objects can be defined: a square, circle and triangle. Markings (thin lines with arrowheads) define which object is created and where it is placed. Object type is defined by the direction of the marking stroke. Objects are centered on the starting point of the defining mark.

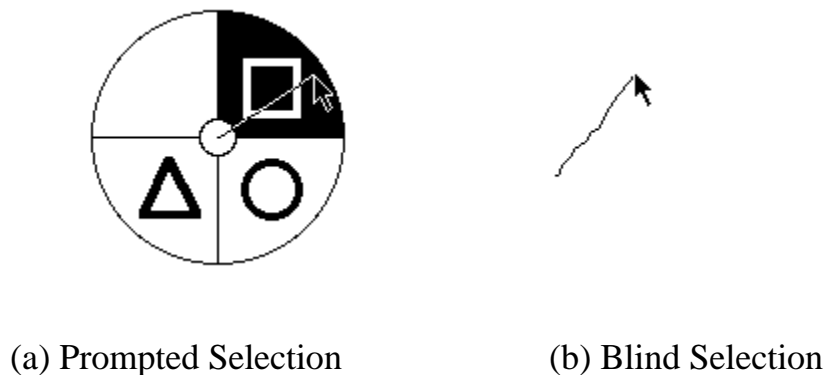
This work is a part of a larger effort to design interfaces which support smooth transfer of skills as novices become expert. Typically, interfaces provide two modes of operation. The first mode, designed for novice users, is self-revealing. Conventional menu-driven interactions are an example of this. The self-revealing component of this mode is emphasized over efficiency of interaction since novice users are more concerned with how to do things rather than how quickly they can be done. The second mode, designed for expert users, typically allows terse, non-prompted interactions. Command line interfaces and accelerator keys are examples of this mode. Usually there is a dramatic difference between novice and expert behavior at the level of physical action (for example, a novice uses the mouse to select from a menu whereas an expert presses an accelerator key). It is our goal to reduce this discrepancy in action without reducing efficiency of the expert and ease of learning for the novice. When the basic action of the novice and expert are the same, novice performance builds the skills that lead to expert performance in a smooth and direct manner.

### 1.1 A Mechanism for Self-Revealing Markings

We suggest that "pie menus" (Callahan et al., 1988) can be used to make marking self-revealing. Pie menus can be used to reveal what options are available to the user, as well as the mapping between marking and command. In our interface, if a user is unsure of what markings can be made, the user presses down on the stylus and waits for a short interval of time (approximately 1 second). In this case, when the system detects that no mark is being made it prompts the user with a pie menu of the available commands, these appearing directly under the cursor.[\[1\]](#) The user may then select a command from the pie menu by keeping the stylus tip depressed and making a stroke through the desired sector or slice of the pie. The slice is highlighted and the selection is confirmed when the pressure on the stylus is released. A user can also indicate "no selection" by moving the stylus tip back to the center of the menu before releasing, or change selection by moving the tip to highlight another slice before

releasing.

The first important point to note is that *the physical movement involved in selecting a command is identical to the physical movement required to make the mark corresponding to that command*. For example, a command that requires an up and to the right movement for selection from the pie menu, requires an up and to the right marking in order to invoke that command. Note the concept is similar to that of accelerator keys in many of today's applications. A user is reminded of the keystrokes associated with particular menu items every time a menu is displayed since the name of the corresponding keys may appear next to the commands. The difference is that with our marking/pie menu mechanism, the user is not only reminded, but actually rehearses the physical movement involved in making the mark every time a selection from the menu is made. We believe that this further enhances the association between mark and command.



**Figure 2.** *The transition from novice to expert reflected in two different ways of invoking commands.*

The second point to note is that *supporting markings with pie menus in this way helps users make a smooth transition from novice to expert*. Novices in effect perform "menu selection". We have observed in the laboratory that users almost always wait for the pop-up menu and then select the desired sector (Figure 2a) when they first encounter a new menu. However, waiting for the menu takes time, and thus as users begin to memorize the layout (as they become expert), they begin to "mark ahead" to invoke the commands instead (Figure 2b). We have also observed an intermediate stage where users may begin to make a mark, and then wait for the menu to pop-up in order to verify their choice of action.

This mechanism could be valuable for supporting fast performance on keyboardless computers. Without a keyboard, an expert user is limited to making pointer-driven menu selections since accelerator keys are not available. If a pointing device like a stylus is used, our marking mechanism could replace the role of accelerator keys. A small set of short, straight marks could be associated with the most frequently used commands that do not otherwise have any obvious mnemonic marks to associate with them.

There are three advantages in associating short, straight marks for frequently used commands. First, the frequent use of the mark/menu will reinforce the association between mark and command: some marks can be remembered because they are mnemonic but short straight marks only can be remembered if they are used often. Second, reducing the articulation time of frequently used commands will produce more overall time savings than reducing the articulation time of rarely used commands. Third, computer recognition of straight marks can be very reliable and fast.

## **2. THE EXPERIMENT**

To date there is little research on the use of pie menus in human-computer interaction. Callahan, Hopkins, Weiser and Shneiderman (1988) investigated target seek time and error rates for 8-item menus, but concentrated on comparing them to linear menus. In particular they were interested in what kind of information is best represented in pie menu format. Given our intention of exploiting the marking/pie menu mechanism, there are a range of other issues with regard to both articulatory and cognitive aspects of using such menus that need to be investigated.

We designed an experiment where we systematically varied menu size and input device for three groups of subjects. "Menu size" in this context refers to the number of items in a menu and not to the diameter of the menus. (All menus were of equal diameters.) One group selected target items from fully visible or "exposed" menus (Exposed group). Since there is little cognitive load involved in finding the target item from menus which are always present, we felt that this group would reveal differences across levels of the factors for articulatory aspects of performance. Two other groups selected items from menus which were not visible ("hidden" menus): one with an ink trail (Marking group), and one without (Hidden group). The two hidden menu groups were intended to uncover cognitive aspects of performance since they both involve the added cognitive load of remembering the location of the target item. Comparing use of an ink trail with no ink trail was intended to reveal the extent to which supporting the metaphor of marking and providing added visual feedback affects performance.

We formed the following specific hypotheses:

1. Exposed menus will yield faster response times and lower error rates than the two hidden menu groups. However, performance for the two hidden groups will be similar to the Exposed group when menu size is small. When menu size is large, there will be greater differences in performance for hidden versus exposed menus. This will be due to the difficulty of remembering and mentally representing large hidden menus.
2. For exposed menus, response time and number of errors will monotonically increase as menu size increases. This is because performance on exposed menus is mainly limited by the ease of articulation of menu selection, as opposed to ease of remembering or inferring the menu layout.
3. For hidden menus (Marking and Hidden groups), response time will not be solely a function of menu size. Instead, menu layouts that are easily inferred or that are familiar will tend to facilitate performance (such as sizes 8 and 12). Menu sizes that are more difficult to remember or mentally construct (such as 7 and 11) will tend to degrade performance.
4. The stylus will outperform the mouse both in terms of response time and errors. The mouse will outperform the trackball. This prediction is based on previous work (Mackenzie, Sellen, and Buxton, 1991) comparing these devices in a Fitts' law task.
5. Differences in performance due to device will not depend on whether the menus are hidden or exposed, or whether or not markings are used. The rationale for this is that we assume device differences are mostly a function of articulation rather than originating from the cognitive level.
6. Users will make essentially straight strokes when selecting from menus but straight strokes will be particularly evident in the Marking group. This is because the visual feedback provided in the Marking group and the fact that the menus are hidden, supports the "marking" metaphor as opposed to the "menu selection" metaphor.
7. Performance on hidden menus (Marking and Hidden groups) will improve steadily across trials. Performance with exposed menus will remain fairly constant across trials. This prediction is based on the fact that we believe articulation of selection does not greatly improve with practice, but that learning the menu layouts does.

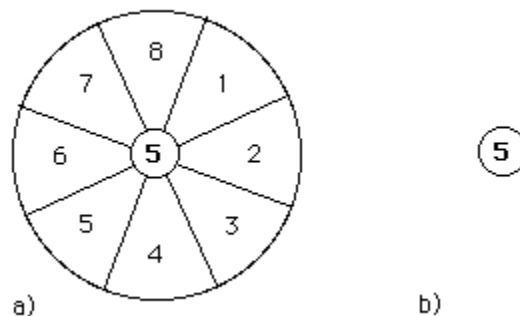
## **2.1 Method**

**Subjects.** Thirty-six right-handed subjects were randomly assigned to one of three groups (Exposed, Hidden, and Marking groups). All but one had considerable

experience using a mouse. Only one subject had experience using the trackball. None of the subjects had experience with the stylus.

**Equipment.** The task was performed on a Macintosh IIX computer. The mouse used was a standard Macintosh mouse set to the smallest control/display ratio. The trackball used was a Kensington TurboMouse, also set to the smallest control/display ratio. The stylus was a Wacom tablet and pressure-sensitive stylus (an absolute device). The control/display ratio used was approximately one-to-one.

**Task.** Subjects used each of three input devices to select target "slices" from a series of different sizes of pie menus as quickly and as accurately as possible. The pies contained either 4, 5, 7, 8, 11, or 12 slices. All pie menus contained numbered segments, always beginning with a "1" immediately adjacent and to the right of the top segment. The other slices were labelled in clockwise order with the maximum number at the top (see Figure 3a). The diameter of all pie menus was 6.5 cm., and Geneva 14 point bold font was used to label the slices.



**Figure 3.** An eight slice pie menu in the Exposed group (a) and in the Hidden or Marking group (b).

In designing this experiment, a great deal of time was spent discussing what kind of items should be displayed in the pie menus. Menus in real computer applications usually contain meaningful items, but the order in which they appear is not easily inferred. The numbered menus we used, on the other hand, used ordered, meaningless labels. We reasoned that we were more interested in how easily users could articulate the actions, and mentally represent or construct menu layout, given that they knew both the contents of the menus and the order in which they appeared. Meaningful items randomly ordered would require more learning time in order to memorize the layout. However, as Callahan et al. (1988) have shown, performance varies widely depending

on the kinds of items represented. Thus, learning time would be much longer, without being easily generalizable to any particular interface. By using ordered but meaningless items, we could focus more quickly on the factors of interest: menu size, input device, the difference between exposed and hidden menus, and the use of an ink trail.

In the *Exposed* menu group, the entire menu was presented on each trial (Figure 3a). The target number corresponding to the slice to be selected was presented when the subject located the cursor within the center circle of the pie menu and either pressed down and held the mouse or trackball button, or pressed down and maintained pressure on the stylus. The subject's task was then to maintain pressure and move in the direction of the target slice, which would highlight as the cursor moved over it. Releasing the button, or pressure, confirmed selection of the slice. The slice would remain highlighted even if the cursor went outside the outer perimeter of the pie menu, and confirmation of selection could occur by releasing outside the menu itself. Subjects could also change their selection by moving the cursor around the menu to highlight other segments, as long as pressure on the button or stylus was maintained. After the selection was confirmed, the menu would "gray out" displaying the menu with the slice selected for a period of 1 second. If an incorrect segment was selected, the Macintosh would beep on release.

In the *Hidden* menu group, the task was essentially the same, except that during selection, only the central circle of the pie menu would be visible (Figure 3b). After confirming the selection, subjects would receive the same grayed out feedback as in the Exposed group, indicating which response had been made, and whether or not it had been in error. The *Marking* group was almost identical to the Hidden group, except that the movement of the cursor with the button depressed left an "ink trail".

After each trial, subjects received a running score, presented in the lower right-hand corner of the screen. A minimum of 10 points could be obtained for each correct response, with more points scored as response time became shorter. However, subjects were penalized 20 points for errors. At the end of each set of block of trials, each subject's current performance was shown in relation to the best score obtained by other subjects in the same conditions. The scoring criterion was the same for all groups.

***Design and Procedure.*** The experimental design consisted of one between-subjects factor (group) with three levels, and two within-subjects factors (device and menu size) with three and six levels, respectively. The main dependent variables of interest were response time and number of errors. Response time was defined as the total time

from presentation of the target number to confirmation of the selection. An error was defined as an incorrect selection.

One third of the subjects was randomly assigned to the Exposed group, one third to the Hidden group, and one third to the Markings group. Every subject used each of three input devices (mouse, trackball, and stylus). Trials were blocked by device, with the order of device determined by a 3 X 3 Latin Square.

For each device, subjects *in all groups* began by practicing on exposed menus for a total of six trials for each menu size (4, 5, 7, 8, 11 and 12). During practice, menu size was blocked and presented in random order. This practice period was intended to acquaint subjects with the feel of the particular input device they were about to use. It also provided an opportunity for subjects to familiarize themselves with the layout of the menus before beginning the timed trials.

Subjects in the Exposed group then moved on to the timed trials, while subjects in the Hidden and Markings groups received a further set of practice trials designed to acquaint them with the "feel" of hidden menus. For this practice session, menu sizes of 3 and 6 were used (six trials each) since these menu sizes were not used in the actual timed trials. This was a deliberate attempt to equalize exposure to the menus of interest in the three groups.

For the timed portion of the experiment, trials were blocked by menu size (4, 5, 7, 8, 11, or 12). Menu size was randomly permuted for each subject. Each subject began a particular block by first studying the menu layout for 6 seconds. They then received a total of 40 trials for each menu size with a short break at intervals of 10 trials. Targets were drawn randomly from a uniform distribution with replacement, with the added constraint that no target could be repeated on consecutive trials.

## 2.2 Results and Discussion

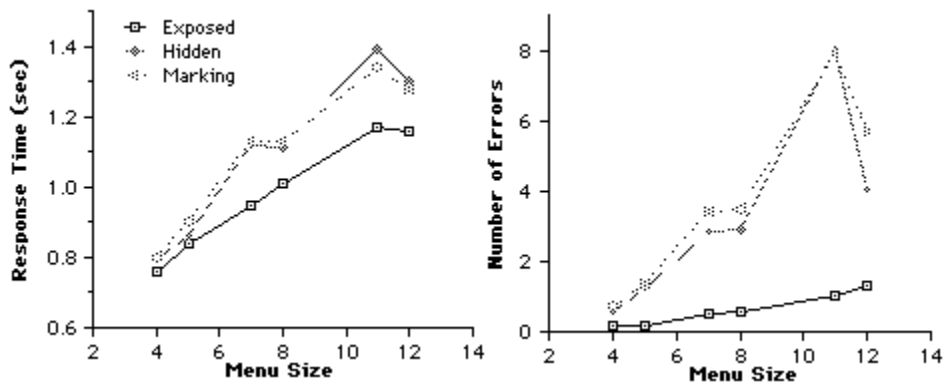
<b>Group</b>	<b>Mean RT (SD) (sec.)</b>	<b>Mean Number of Errors in 40 Trials (SD)</b>	<b>Mean Percentage Errors</b>
<b>Exposed</b>	0.98 (0.23)	0.64 (1.00)	1.6%
<b>Hidden</b>	1.10 (0.31)	3.27 (3.57)	8.2%
<b>Marking</b>	1.10 (0.31)	3.76 (3.67)	9.4%

**TABLE 1:** Mean response times (sec.), mean number of errors and mean percentage of errors per group.

The average total response times and mean number of errors, and mean percentage of errors per group are shown in Table 1. Response times were calculated after eliminating trials over 4 seconds. On close inspection of the data, we decided that trials with abnormally long response times reflected periods of excessive loss of concentration, confusion or distraction. However, this criterion resulted in disqualification of only 0.4% of the total responses. In addition, we based response time on correct responses only, since error trials were likely to add unwanted variance to the data.

### Menu Size Effects

As expected, increasing the menu size significantly increased response time ( $F(5,55) = 388.4, p < .001$ ), and errors ( $F(5,55) = 382.8, p < .001$ ). In addition, there were overall performance differences among the groups in terms of errors ( $F(2,22) = 21.97, p < .001$ ) but not in terms of response time. However, these main effects are difficult to interpret because differences among groups depend on menu size (see Figure 4). That is, there was a significant interaction between group and menu size both in terms of response time ( $F(10,110) = 3.5, p < .001$ ) and errors ( $F(10,110) = 64.7, p < .001$ ).



**Figure 4.** Response time and average number of errors (of a total of 40 trials) as a function of menu size and group.

These results address the first three hypotheses:

- (1) In general, as predicted by Hypothesis 1, exposed menus tended to yield faster and more accurate performance than hidden menus, both with and without markings. Mean response time is consistently lower in the Exposed group versus the Hidden and

Marking groups. In the case of errors, collapsing across menu size yields significant group differences. Post hoc comparisons of error means (Tukey HSD,  $\alpha = .05$ ) show no differences between the two hidden groups, but both produced significantly more errors than the Exposed menu group.

In addition, as also hypothesized, performance on hidden menus tends to converge with performance on exposed menus when the menu size is small. This is supported by the significant group by menu interactions reported above. To further show the convergence, specific comparisons were carried out to test for differences between performance on menu sizes 4 and 5 for the hidden groups versus the Exposed group. No difference was found between the hidden groups and the Exposed group for menu size 4. For menu size 5, there was also no difference between hidden and exposed menus in terms of errors, although the response times were significantly different ( $F(1,110) = 6.5$ ). When we consider that this analysis collapses across trials, the case for no difference is made stronger as it takes into account the large differences that may exist before much learning has taken place.

(2) Our second hypothesis predicted that in the Exposed group, response time and errors would monotonically increase as a function of menu size. In the case of errors, this relationship seems to hold. However, this must be qualified by the fact that errors were infrequent and thus floor effects may obscure the true shape of the function.

Response time also increased monotonically except for menu size 12. Specific comparisons at the .05 level confirm significant increases in response time from menu size 4 to 5 ( $F(1,55) = 16.8$ ), and from size 7 to 8 ( $F(1,55) = 7.4$ ), but that there are no differences between menu sizes 11 and 12.

There are at least two possible reasons why response time does not increase from menu size 11 to 12. First, it could be a case of diminishing effects. Adding an extra item to a menu of size 4 represents a 20% increase in the number of items, whereas, adding an extra menu item to a menu of size 11 represents only an 8% increase in number of items. Another possibility is that familiarity with the "clock face" layout may reduce the time for visual search and thereby speed performance.

(3) The pattern of results predicted by Hypothesis 3 is also supported: when menus were hidden, some menu sizes were easier to evoke or reconstruct from memory than others. This was not purely a function of menu size. Note first that the patterns of performance for the two hidden menu groups (Hidden and Marking) are very similar to each other and that the curves for response time show a similar pattern to those for

errors. The latter suggests that the two indices of task difficulty are highly correlated with each other.

The characteristic curve that emerges (Figure 4) shows that performance in general does tend to degrade as menu size increases, but that certain menu sizes do not follow this pattern (i.e., sizes 8 and 12). This is confirmed by a series of specific comparisons. No response time differences were found in either group between performance on menu size 7 versus 8. Further, performance on menu size 12 is faster than menu size 11 for the Hidden group ( $F(1,55) = 11.25$ ) but does not reach significance for the Marking group. In terms of accuracy, the only significant difference was again consistent across groups with pie size 12 more accurate than size 11 (Hidden,  $F(1,55) = 50.96$ ; Marking  $F(1,55) = 13.51$ ). By contrast, for both groups, tests show that on the consecutive menu sizes 4 and 5 performance on menu size 4 is faster than on menu size 5 (Hidden,  $F(1,55) = 4.05$ ; Marking  $F(1,55) = 9.00$ ).

The results show that menu size 12 in particular facilitates performance. Many subjects mentioned that the metaphor of a clock face helped them to select the target item because it could be brought readily to mind. Thus it seems reasonable to suggest that it is the cognitive bottleneck, or the difficulty of evoking the mapping between target and action that limits performance.

## **Device Effects**

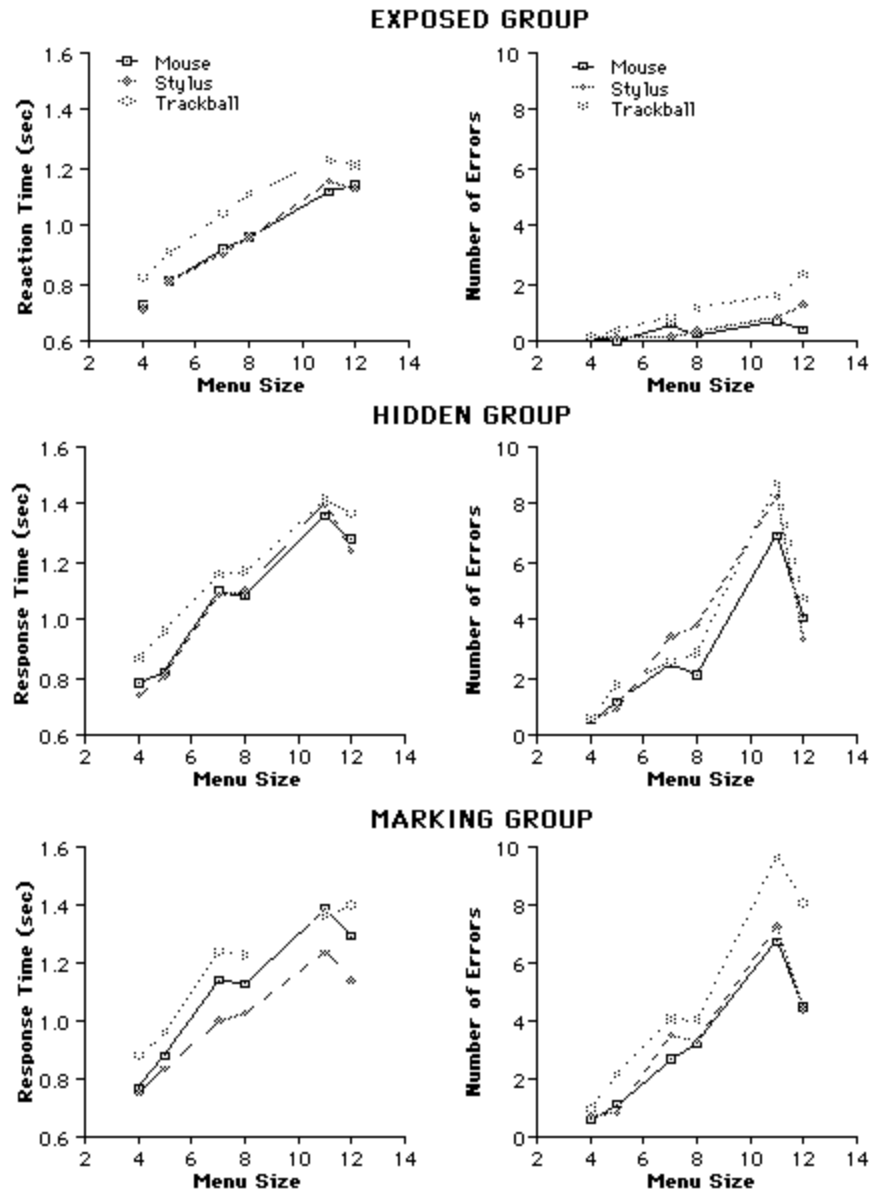
As predicted by Hypothesis 4, stylus and mouse outperformed the trackball. There was a significant main effect of input device for response time ( $F(2,22) = 9.64$ ,  $p < .001$ ) and errors ( $F(2,22) = 11.29$ ,  $p < .001$ ). Pairwise comparisons (Tukey HSD test,  $\alpha = .05$ ) showed the trackball was both significantly slower and gave rise to more errors than the stylus or mouse. However, contrary to our expectations, there was no difference in mean response time or errors, between the stylus and mouse.

Initial analyses appeared to support our fifth hypothesis in which we predicted that the effect of input device would not depend on whether or not the menus were exposed, or whether or not there was an ink trail. Input device did not interact with group, either in terms of response time or errors. [2] However, on closer examination, a more interesting result emerged.

We discovered that in the Marking group, the stylus was significantly faster than both the trackball and mouse, with no difference between the trackball and mouse. In the Exposed group, the trackball was slower than both the mouse and stylus, with no difference between the mouse and stylus. These two discoveries were based on separate analyses of variance for each of the three groups on the response time data.

There were significant differences among devices in the Exposed ( $F(2,22) = 10.44, p < .001$ ) and Marking groups ( $F(2,22) = 8.32, p < .002$ ), but not in the Hidden group. Tukey tests ( $\alpha = .05$ ) were used to more closely examine these differences and revealed the superiority of the stylus in the Marking group and the inferiority of the trackball in the Exposed group. No significant device by menu size interactions were found in any of the three groups. The data are shown graphically in Figure 5.

There may be two reasons for the superiority of the stylus when markings are added to selection from hidden menus. First, it is often difficult to perceive when enough pressure is being applied to the stylus to make a selection. Thus, providing visual feedback when this state is maintained may be important to realize the full potential of this device. Second, providing an ink trail is consistent with the metaphor of marking with a pen, which may improve performance. Alternatively, failing to support the pen metaphor by not providing the ink trail (Hidden group) may violate users' expectations and thus negatively affect performance.



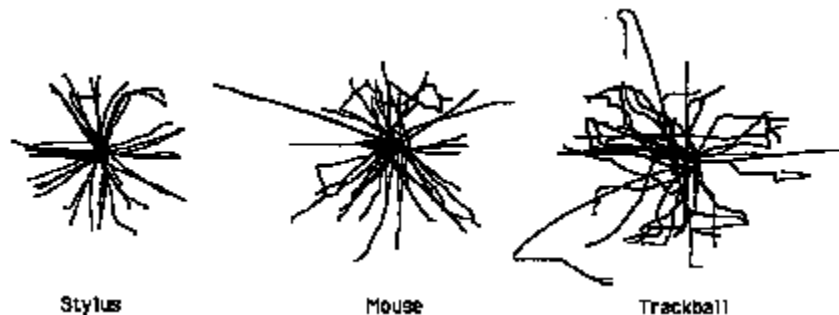
**Figure 5.** Response time (sec.) and average number of errors (out of 40 trials) as a function of device, menu size, and group.

Separate analyses of the error data within each group further supported the inferiority of the trackball in pie menu selection. The trackball was found to be the source of significant device differences in the Exposed ( $F(2,22) = 9.92, p < .001$ ) [3] and Marking groups ( $F(2,22) = 9.92, p < .001$ ). Pairwise comparisons (Tukey,  $\alpha = .05$ ) in the Exposed and Marking groups showed differences between the trackball and the other two devices, and no difference between mouse and stylus.

The finding that the trackball was no worse than the mouse and stylus in the Hidden group may be due to the fact that in both the Exposed and Marking groups, visual feedback emphasized the difficulty of articulating the actions of the trackball thereby causing performance to be worse. In the Exposed case, sectors were highlighted as they were selected and it is possible that the trackball caused a great deal of reselection. In the Marking case, users reported that the ink trail was disturbing in conjunction with the trackball because the paths looked erratic and inaccurate.

## Stroke Analysis

We were interested in seeing if subjects used straight strokes when making selections. This was important to discover, because if menu selection tended to be done in some manner other than a straight stroke, we could not claim that users rehearse the physical movement required to make the corresponding marks. Thus we would not expect as much transfer of skill between making menu selections and making markings. Further motivation is for technical purposes. Unlike conventional menu selection which is based only on the last location of the cursor, mark recognition systems take the entire shape of the stroke into account. Therefore the success of recognition depends to some extent on knowing the shapes of the strokes that users tend to create.



**Figure 6.** *The marks a subject used in selecting from a hidden 12 slice menu.*

To address these issues we recorded and displayed the path data for users' individual strokes. Figure 6 shows the complete set of all strokes made by one subject in the Hidden group when presented with a 12-slice menu. These strokes are typical of the strokes made by other subjects in all groups.

In general, subjects used approximately straight strokes. No alternate strategies such as always starting at the top item and then moving to the correct item were observed. However, there was evidence of reselection from time to time, where subjects would

begin a straight stroke and then change stroke direction in order to select something different.

Surprisingly, we observed reselection even in the hidden menu groups. This was especially unexpected in the Marking group since we felt the affordances of marking do not naturally suggest the possibility of reselection. It was clear though, that training the subjects in the hidden groups on exposed menus first made the option of reselection apparent. Clearly many of the subjects in the Marking group were not thinking of the task as making marks *per se*, but of making selections from menus that they had to imagine. This brings into question our *a priori* assumption that the Marking group was using a marking metaphor, while the Hidden group was using a menu selection metaphor. This may explain why very few behavioral differences were found between the two groups.

Reselection in the hidden groups most likely occurred when subjects began a selection in error but detected and corrected the error before confirming the selection. This was even observed in the "easy" 4-slice menu, which supports the assumption that many of these reselections are due to detected mental slips as opposed to problems in articulation. There was also evidence of fine tuning in the hidden cases, where subjects first moved directly to an approximate area of the screen, and then appeared to adjust between two adjacent sectors.

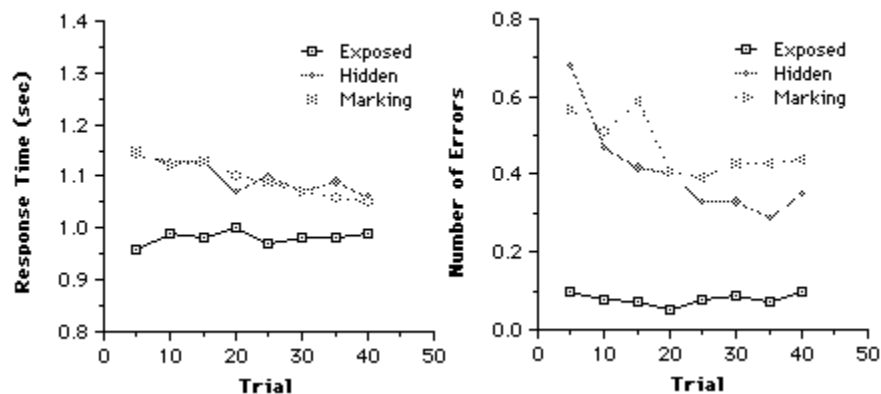
Other observations on the basis of this analysis include individual differences in stroke length. Average stroke length varied by as much as a factor of two between subjects. In addition, across all groups, menu sizes, and subjects, strokes produced with the trackball appeared more jagged and less controlled than those made with the mouse or stylus. This is consistent with the statistical results showing that the trackball tends to be slower and less accurate than stylus or mouse. For menu size 4, most subjects showed straighter marks with the stylus as opposed to the mouse. The presence or absence of an ink trail did not appear to make any discernable difference to stroke shape.

## **Learning Effects**

The forty trials for each menu size were divided into 8 consecutive blocks. Response time and mean errors were calculated for each 5-trial block in order to look more closely at learning effects. Overall, there was a small but steady decrease in response time over trials which was statistically significant ( $F(7,77) = 5.79, p < .001$ ). With regard to errors, a significant main effect of trial was also found ( $F(7,77) = 10.52, p < .001$ ).

We have argued with the support of the data that the performance limiting factor for exposed menus is the difficulty of articulating selection actions. In the case of the two groups using hidden menus, the data support the claim that the factor limiting performance is cognitive. In other words, the time it takes to remember or infer the correct mental representation becomes the overriding factor determining performance. Thus, performance in the Exposed group can serve as a baseline measure that users should approach as they become expert.

Hypothesis 7 states that the cognitive component is the one that is most affected by learning, as opposed to the articulatory component. Thus, we expect a steady improvement in performance in the two hidden groups, as opposed to fairly constant performance in the Exposed group over time.



**Figure 7.** Group effects in terms of response time and number of errors in 5 trial intervals.

As is shown in Figure 7, response time in the hidden groups appears to improve across trials while the curve for the Exposed group is fairly flat. Errors also remain relatively constant for the Exposed group over trials, while decreasing on average for the two hidden groups. Support for Hypothesis 7 is found in a significant group by trial interaction for response time ( $F(14,154) = 2.90, p < .001$ ) and errors ( $F(14,154) = 3.15, p < .001$ ).

As a final point, it follows from the above reasoning that we would expect no significant interaction of input device by trial, since type of input device would presumably have the greatest impact on the articulation as opposed to the cognitive component of performance. The fact that no significant interaction of device by trial for was found is consistent with expectation.

### 3. SUMMARY AND GENERAL DISCUSSION

The results and their implications for design can be summarized as follows:

**Result 1.** As predicted, hiding the pie menus both slows performance and increases the error rate, especially for large menu sizes. As menu size increases, added to the problems of articulation is the difficulty of successfully mentally reconstructing the menu layout or remembering the necessary strokes to make menu selections. However, when menu size is small (up to 5 slices), there is little or no performance difference, even early in practice.

**Design Implications.** For ordered sets of commands, users should be as fast and error-free in making markings as in selecting from a visible pie menu of up to 5 slices. If the commands are not ordered, then it may take more time to acquire the skill. However, command semantics can be exploited. For example, "Open" and "Close" can be positioned opposite to each other, as can "Cut" and "Paste". This may speed the learning process and allow users to mark ahead faster. In addition, the most frequently used commands form a very small set, and thus we can be optimistic that these can be invoked successfully through the hidden pie menu mechanism.

**Results 2 and 3.** For exposed menus, the results showed performance declines steadily as menu size increases. This is probably due to two factors: (1) the increasing reaction time to visually search and choose among alternatives, and (2), the increasing difficulty of articulating the action as targets become smaller.

The pattern for hidden menus was different however. Instead of a monotonically increasing response time and error rate as a function of menu size, certain menu sizes facilitated performance, while others were particularly difficult. Specifically, even-numbered menu sizes allowed subjects to easily find the target slice (sizes 4, 8, and 12). For example, a 12-slice menu facilitated performance probably by association with the clock-face metaphor. Subjects reported that the 8-slice menu was easy to learn because they could easily mentally subdivide the pie and infer the position of the target slice. On the contrary, menu sizes 5, 7, and 11 presented subjects with more difficulty.

**Design Implications.** When menus are hidden, overcoming the "cognitive bottleneck" or the difficulty of learning and using mental representations of menus can be facilitated by using layouts which exploit known metaphors, or which are easily subdivided. Using an even number of items or laying out items at the points of a compass or hour positions of clock can be used to counteract the increased difficulty of large menu sizes. The ease with which subjects acquired and performed with the

12-slice menu is testimony to the strength of a good metaphor. One could imagine a user remembering a command location or marking by mapping it to an hour/hand position: "undo is at 3 o'clock".

**Results 4 and 5.** The stylus and mouse outperformed the trackball both in terms of response time and errors. Analysis of the paths showed that paths made with the trackball were more jagged and less controlled than those made with the mouse or stylus. The stylus and mouse yielded similar performance, with the exception that the stylus was significantly faster than the mouse when an ink trail was present.

**Design Implications.** The results speak strongly against using a trackball for the marking/pie menu mechanism. Further, subjects' comments suggest that the combination of trackball and ink trail was especially bad. One subject complained of being disturbed by the messy ink trail left when using a trackball. It seems that the visual feedback provided by the ink trail only served to emphasize the inadequacy of the paths made by this device.

The general performance similarity of the mouse and stylus suggests that either may be appropriate devices for this kind of mechanism. Two cautionary notes should be made, however. First, it is likely that the ink trail added important feedback to tell the user when the appropriate amount of pressure was being applied to the stylus. This suggests that another kind of stylus (i.e. one with audio or tactile feedback to indicate a "button-click") might have fared better against the mouse in all groups. It also reveals a design deficiency of the stylus that could easily be overcome. Second, while the mouse and stylus yielded similar performance, observation of people using the mouse to make marks other than straight strokes suggests that the mouse may be inferior to the stylus in other situations. For example, an early version of the GEdit application described in the introduction required users to make an upside-down "V" to create triangles. This turned out to be particularly difficult to accomplish with a mouse, but not with a stylus. Thus, if the interface uses marks other than straight strokes, a stylus seems to be the best choice.

**Result 6.** Subjects used essentially straight strokes. However, there was evidence of reselection (where subjects would begin a straight stroke and then change stroke direction in order to select something different) even in the hidden groups. This casts doubt on our initial assumption that subjects in the Marking group would begin to think of the task as making marks, instead of making menu selections. Instead, it suggests that they thought of the task in terms of making selections from the exposed menus they were trained on, which now happened to be hidden. Markings themselves do not afford reselection, whereas the mechanism of the pie menus does.

The fact that the marking metaphor was not supported as strongly as we had hoped may account for the fact that no major differences were found between the Hidden and Marking groups. For example, the presence or absence of an ink trail did not appear to make any discernable difference to stroke shape.

***Design Implications.*** Since users tended to make straight strokes we are optimistic that users are rehearsing the physical movement required to make marks as they perform menu selection. This bodes well for the transfer of skill from novice to expert. Also, since reselection in the Marking group occurred infrequently, this indicates that even if users think of the task not as making marks but as selection from hidden menus, mark recognition failure due to non-straight reselection strokes will be infrequent.

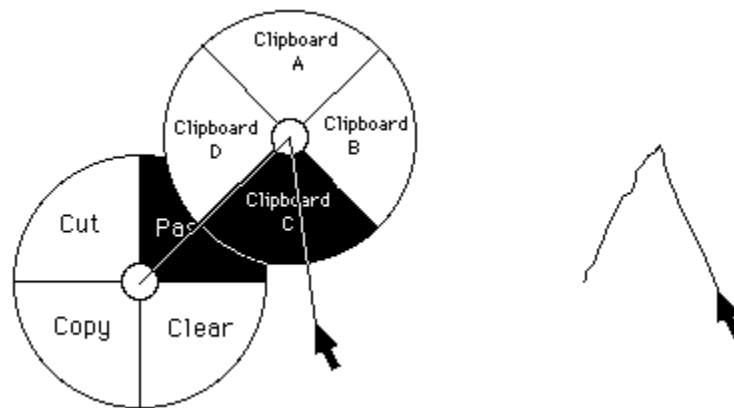
***Result 7.*** Performance across trials was uniform for exposed menus but underwent steady and significant improvement across trials for hidden menus (both groups). We argue that the performance limiting factor for exposed menus is the difficulty of articulating selection actions, whereas in the hidden groups the limiting factor is the time it takes to evoke or construct the correct mental representation. Articulation skills were acquired fairly rapidly and reached stable performance. Thus performance in the Exposed group provides a baseline measure that users of hidden menus approach.

***Design Implications.*** The substantial improvement for hidden menus over only 40 trials suggests that if the menus contain meaningful and frequently used commands, users will acquire the necessary skills quickly and easily. Both response time and error rates can be expected to rapidly improve with time. The question of *how much* practice is necessary for hidden menu performance to equal exposed menu performance, and how that varies with menu size is an issue for further research and analysis. Meanwhile, we can be confident that small menu sizes will enable users to quickly begin marking ahead. It will be important to provide the pop-up menu mechanism for larger menu sizes.

#### **4. CONCLUSIONS AND FUTURE DIRECTIONS**

We have demonstrated the basic concepts on which the marking/pie menu mechanism is based. For example, users are able to "mark ahead" and to do so quite successfully especially when menus are small. The ability to mark ahead in a sense puts the user more in control, in that no prompting by the system is explicitly forced on the user. In addition, using marks has the benefit that no time is spent waiting for a menu to be displayed or acquiring the menu (as in pull-down menus or margin menus). Further, a mark is less of a visual disturbance than a pop-up menu which obliterates part of the screen.

We have also experimentally explored a number of the articulatory and cognitive aspects in using such a mechanism, and discussed the design implications of the findings. One interesting extension of this work involves the design, implementation, and behavioral aspects of hierarchical or nested pie menus (see Figure 8). Similar to the mechanism proposed in this study, prompted selection from a hierarchical pie menu results in a unique stroke pattern. This stroke pattern defines the shape of the marking required to blindly select an item from the menu hierarchy. Selection confirmation, as one moves through the hierarchy, is also possible. It would be interesting to investigate the feasibility of this mechanism and its performance relative to non-hierarchical menus.



**Figure 8.** Selection from a hierarchical pie menu and the corresponding mark.

There are other mechanisms besides pie menus which can make markings self-revealing. Pie menus are a specific case of a class of interface mechanism which prompts the user to make the physical motion required for the corresponding mark. Another possibility is that of a "donut menu" [4] -- a menu divided into concentric circles rather than sectors, where each concentric circle corresponds to a different command. The corresponding marks are therefore discriminated by length rather than angle. It would be interesting to evaluate such alternate prompting mechanisms.

Despite the value of such controlled studies, there are a number of questions which can only be answered by careful design and implementation of this mechanism in a real application. How often will users wait for a pop-up menu? What are the issues involved in integrating such a mechanism into a larger more complex interface? The next logical step is to perform a case study where users' performance is tracked in the context of a useful application and real tasks.

**Acknowledgements.** We thank the members of the Input Research Group at the University of Toronto who provided the forum for the design and execution of this project. In particular we thank Scott MacKenzie for many useful comments on this paper and Rich Helms of IBM Canada Laboratory for his design suggestions. We also thank Richard Young of the MRC Applied Psychology Unit, Cambridge, for helpful comments on the first draft of this paper. This work was performed in the Dynamic Graphics Project laboratory at the University of Toronto, to whom we are grateful. We especially thank Alison Lee for her advice on data visualization.

**Support.** We gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada, Digital Equipment Corporation, and Xerox PARC.

## REFERENCES

- Callahan, J., Hopkins, D., Weiser, M. & Shneiderman, B. (1988). An empirical comparison of pie vs. linear menus. *Proceedings of CHI '88*, pp. 95-100.
- Carr, R.M. (1991). The point of the pen. *Byte*, Vol. 16(2), pp. 211-221.
- Hardock, G. (1991). Design issues for line driven text editing/annotation systems. *Proceedings of Graphics Interface '91*, Calgary, June, 1991.
- Kurtenbach, G. & Buxton, W. (1991). GEdit: A testbed for editing by contiguous gesture. *SIGCHI Bulletin*. pp. 22-26.
- Leitch, C. (1990). High-tech pen: Big deal or bust. *Toronto Globe and Mail, Report on Business*, pages C1 & C11, Oct. 23, 1990.
- Mackenzie, I. S., Sellen, A. J., and Buxton, W. (1991). A comparison of input devices in elemental pointing and dragging tasks. *Proceedings of SIGCHI '91*, New Orleans, LA. New York: ACM, pp. 161-166.
- Normile, D. & Johnson, J.T. (1990). Computers without keys. *Popular Science*, August, 66-69.
- Rebello, K. (1990). New PCs can kiss keyboards goodbye. *USA Today*, p. 6B, Feb. 22.
- Rubine, D. H. (1990). *The automatic recognition of gestures*. Unpublished doctoral thesis, Dept. of Computer Science, Carnegie Mellon University.

Welbourn, L. K. & Whitrow, R. J. (1988). A gesture based text editor. *People and Computers IV, Proceedings of the Fourth Conference of the British Computer Society Human-Computer Specialist Group*. Cambridge, Cambridge University Press, pp. 363-371.

Wolf, C. G., Rhyne, J. R. & Ellozy, H. A., (1989). The paper-like interface. *Designing on Using Human Computer Interface and Knowledge-Based Systems*. Amsterdam: Elsevier Science Publisher, pp. 494-501