

## Programmation des Interfaces Interactives Avancées

### Travaux pratique numéro 2 - Correction

#### Exercice 1. [inspiré de M. Baudoin-Lafon]

Les temps empiriques de pointage avec un périphérique d'entrée sont donnés dans le tableau suivant.

| D \ L | 1.00 | 3.00 | 6.00 |
|-------|------|------|------|
| 0.10  | 0.50 | 0.67 | 0.77 |
| 0.50  | 0.29 | 0.43 | 0.53 |
| 1.00  | 0.23 | 0.34 | 0.43 |

**Tableau 1 : Temps empirique de pointages avec le périphérique d'entrée en fonction de la distance D et de la taille L de la cible. Les distances et tailles sont en cm.**

1) Montrez que ce périphérique suit la loi de Fitts, en déterminant les constantes empiriques de la loi.

*On prend la valeur dans la première colonne, dernière ligne et on a :  $a+b=0.23$*

*On prend la valeur dans la deuxième colonne, dernière ligne et on a :  $a+2b=0.34$*

*D'où  $a=0.12$  et  $b=0.11$*

*$T=0.12 + 0.11*\log_2(D/L+1)=0.12 + 0.159*\ln(D/L+1)$*

2) Évaluez le temps maximal mis pour sélectionner un item dans une hiérarchie de menus à deux niveaux (barre de menu → menu 1 → menu 2). On suppose qu'il y a 6 items dans chaque menu et sous-menu. La largeur d'un item est de 0.5 cm. Les menus se déroulent à la verticale. On suppose que les mouvements horizontaux de la souris sont négligeables. Au début, le curseur est positionné sur la barre de menu.

*La valeur maximale de D est 3cm. En outre  $L=0.5$ , donc le temps de pointage dans chaque niveau de menu est 0.43. Donc pour 2 niveaux, le temps maximale est  $2*0.43=0.86s$*

3) De même, évaluez le temps maximal pour sélectionner un item dans un menu non-hiérarchique (barre menu → menu) ayant 12 items. Même question pour un menu avec 18 puis 24 items.

*Pour 12 items,  $D=6cm$  et  $L=0.5cm$ , donc le temps maximal de pointage est 0.53s.*

*Pour 18 items,  $D=9cm$  et  $L=0.5cm$ , donc le temps maximal de pointage est 0.59s*

*Pour 24 items,  $D=12cm$  et  $L=0.5cm$ , donc le temps max de pointage est 0.63s.*

4) De même, évaluez le temps maximal mis pour sélectionner un item dans une hiérarchie de menu à trois niveaux (barre menu → menu 1 → menu 2 → menu 3). On suppose qu'il y a 6 items par menus.

*La valeur max de D est 3cm, et L=0.5cm. Il y a 3 niveaux, donc le temps max de pointage est  $3*0.43=1.29s$*

5) Concluez.

*Le temps de sélection est plus important dans un menu organisé en niveau hiérarchique.*

## Exercice 2.

Sur la page web ci-dessous, les utilisateurs mettent en moyenne 0.56 secondes pour choisir entre une réservation de train, hôtel, location AVIS etc.

The image shows a promotional banner for SNCF with the text "OFFRE EXCEPTIONNELLE" and "DERNIERS JOURS POUR PROFITER DES BILLETS à 35€ MAX". Below the banner is a navigation menu with options: Train, Hôtel, Location AVIS, Mon Chauffeur, Groupe (10 et +), and Junior & Cie. The main search area includes fields for "Départ : gare, adresse, lieu..." and "Arrivée : gare, adresse, lieu...", a date field "Aller 30/01", and a time field "Départ à 16h". There are also links for "Acheter une carte ou un abonnement" and "Ajouter un retour". A "Rechercher" button is at the bottom of the search area.

L'entreprise souhaite évaluer le temps mis par les internautes pour sélectionner leurs types d'abonnements sur la page web ci-dessous.

The image shows a sidebar menu titled "Cartes & abonnements". At the top of the page, there are buttons for "Me connecter" and a link for "Créer mon compte". The menu items are: TGVMAX, Carte Avantage Jeune, Carte Avantage Week-end, Carte Avantage Senior, Carte Avantage Famille, Carte Liberté, Abonnement Forfait / optiforfait, Carte de fidélité SNCF, and Toutes nos cartes et abonnements.

1) Pouvez-vous proposer un mode de calcul pour évaluer ce temps ?

On applique la loi de Hick-Hyman pour trouver la constante dans le cas de la première page web :

$$0.56 = b \times \log_2(6+1) \approx b \times 2.81 \text{ donc } b \approx 0.2$$

On applique ensuite la même loi pour le choix dans la deuxième page web :

$$T = 0.2 \times \log_2(9+1) = 0.66 \text{ s}$$

- 2) Est-ce qu'il serait avantageux de proposer deux menus successifs, le premier proposant 2 choix (les 4 « Cartes Avantage » versus les autres options) et le second proposant soit les 4 cartes « avantage » explicitement indiquées, soit les autres options (i.e. la carte liberté, la carte fidélité, TGVMAX, Abonnement Forfait / optiforfait, Toutes nos cartes et abonnements) ?

Dans ce cas, il faut compter un premier choix  $t_a = 0.2 \times \log_2(2+1) \approx 0.317$ , puis un second qui peut être soit  $t_{b1} = 0.2 \times \log_2(4+1) \approx 0.464$ , soit  $t_{b2} = 0.2 \times \log_2(5+1) \approx 0.517$ s.

Le temps de choix total serait donc soit  $T1 \approx 0.78$ s, soit  $T2 \approx 0.83$ s.

En outre, il faut compter un geste de pointage supplémentaire. En conclusion, ce n'est pas avantageux.

### Exercice 3.

Le programme vi (« visual editor ») est un éditeur très utilisé sous Unix. Dans vi, les opérations d'édition s'effectuent exclusivement avec le clavier. Par contraste, des éditeurs graphiques tels que Notepad ou Gedit se contrôlent à l'aide du clavier ou de la souris.

1) Utilisez le modèle KLM (Keystroke-Level-Model) pour estimer les temps d'exécution des tâches ci-dessous sous vi d'une part, et Notepad ou Gedit (avec utilisation de la souris) d'autre part. Exprimez les temps d'exécution en fonction de paramètres donnés. Un glossaire sommaire de certaines commandes utiles sous vi est donné ci-dessous. Les valeurs des opérations élémentaires du modèle KLM que vous utiliserez sont les suivantes :  $K=0.28$ s,  $P=1.1$ s,  $H=0.4$ s,  $M=1.2$ s. On suppose qu'un clique souris prend le même temps qu'un « keystroke » (ie  $B=K=0.28$ s). Pour chaque opération, on suppose que l'utilisateur a initialement les mains sur le clavier. On suppose également que vi est initialement en mode commande.

- a) Déplacer le curseur jusqu'à une position localisée à  $n$  lignes et  $m$  colonnes de la position courante du curseur. Les nombres  $n$  et  $m$  n'ont qu'un seul chiffre.

$$Vi : 'n' + '\downarrow' + 'm' + '\rightarrow', \text{ c.à.d. } M+K+K+K+K = 1.2 + 4*0.28 = 2.32s$$

$$Notepad : \text{ homing+pointing+click, i.e. } M+H+P+K = 1.2 + 0.4 + 1.1 + 0.28 = 2.98$$

- b) Copier la ligne courante sur la ligne juste en dessous.

$$Vi : 'y' + 'y' + 'p', \text{ ie } M+3*K=1.2+3*0.28=2.04s$$

$$Notepad \text{ avec deux mains : Homing+Click+Pointing+ 'CTRL+C' + Pointing+click+ 'CTRL+V', ie } M+H+K+P+2K+P+K+2K=1.2+0.4+0.28+1.1+2*0.28+1.1+3*0.28=5.48s$$

- c) Effacer 20 caractères successifs. (facultatif)

$$Vi : '2' + '0' + 'x', \text{ ie } M+3K=1.2+3*0.28=2.04s$$

$$Notepad \text{ avec une main : homing + pointing + click + pointing + homing + 'DELETE', ie } M+H+P+K+P+H+K=1.2+0.4+1.1+0.28+1.1+0.4+0.28=4.76s$$

- d) *Changer la casse de 30 caractères minuscules successifs, c'est-à-dire qu'il faut tous les mettre en majuscule. (facultatif)*

*Vi*: '3' + '0' + '~' + ' ', ie  $M+4K=1.2+4*0.28=2.32s$

*Notepad*:  $30*(\text{'SUPPRIMER'} + \text{'SHIFT'} + \text{'lettre'})$ , ie  $M+30*3*K=1.2+30*3*0.28=26,4s$

Au vu des calculs précédents, des deux éditeurs vi ou Notepad (ou Gedit), lequel vous semble le plus efficace au niveau du temps d'exécution.

vi

2) Considérons l'opération du 1)a) :

- a) Lorsque vous utilisez Notepad (ou Gedit) avec la souris, êtes-vous « conscient » de vous rappeler comment utiliser la souris ? Quel type de mémoire utilisez-vous pour manipuler la souris ?

*Mémoire non-déclarative de type procédurale, car les mouvements avec la souris sont des automatismes.*

- b) Lorsque vous utilisez vi avec la commande (2) du glossaire ci-dessous (un nombre+une touche flèche) pour déplacer le curseur, êtes-vous « conscient » de faire un effort mnésique ? Quel type de mémoire utilisez-vous ?

*Mémoire déclarative de type sémantique si les commandes sont connues, sinon mémoire de travail après avoir consulté le glossaire. La tâche fait également intervenir les fonctions exécutives pour prévoir une stratégie et planifier l'action.*

- c) Au regard des deux questions précédentes, des deux éditeurs vi ou Notepad (ou Gedit), lequel requiert la plus grande charge cognitive à votre avis ?

*Vi requiert plus d'effort cognitif, car il requiert plus d'opérations faisant appel aux fonctions exécutives et à la mémoire déclarative. Par contraste, Notepad s'appuie davantage sur des automatismes.*

3) A partir des conclusions des questions 1) et 2), discutez et comparez brièvement les points forts et faibles de chaque éditeur.

*Bien que vi soit apparemment plus efficace en termes de temps d'exécution selon les résultats du modèle KLM, il requiert davantage de charge cognitive. Le modèle KLM ne permet pas d'estimer la charge cognitive, qui correspond au temps d'acquisition.*

## **Glossaire de commandes vi**

Il y a deux modes sous vi : le mode insertion (tous les caractères tapés sont insérés dans le texte) et le mode commande pour les opérations d'édition. Pour passer du mode insertion au

mode commande, on appuie sur la touche Escape. Pour passer du mode commande au mode insertion, on tape 'a' (insertion après le caractère courant) ou 'i' (insertion avant le caractère courant).

Exemples de commandes d'édition :

- (1) Les touches flèches pour déplacer le curseur.
- (2) Un nombre **n** suivi d'une touche flèche pour déplacer le curseur **n** fois dans la direction de la flèche.
- (3) '0' pour positionner le curseur au début de la ligne
- (4) '\$' pour positionner le curseur en fin de ligne
- (5) 'x' pour supprimer le caractère courant. On peut faire précéder 'x' d'un nombre **n** pour effacer **n** caractères successifs.
- (6) 'dd' pour effacer la ligne courante. On peut faire précéder 'dd' d'un nombre **n** pour effacer **n** lignes successives.
- (7) 'dw' pour effacer le mot suivant. On peut faire précéder 'dw' d'un nombre **n** pour effacer **n** mots successifs.
- (8) 'yy' pour copier la ligne courante.
- (9) 'y+n+w' ('y', puis un nombre **n**, puis 'w') pour copier **n** mots à partir de la position courante du curseur.
- (10) 'y+') ('y' puis ') ) pour copier tout le texte jusqu'à la fin de la ligne.
- (11) 'p' pour coller après le curseur.
- (12) 'P' pour coller avant le curseur.
- (13) '~+' (tilde puis espace) pour changer la casse du caractère courant. On peut faire précéder '~+' d'un nombre **n** pour changer la casse de **n** caractères successifs.