

KeyGlasses : Des touches semi-transparentes pour optimiser la saisie de texte

Mathieu Raynal

Equipe DIAMANT
IRIT UMR CNRS 5505
118, Route de Narbonne
31062, Toulouse Cedex 4, France
raynal@irit.fr

RESUME

Cet article présente un système d'optimisation pour claviers logiciels. Celui-ci repose sur l'ajout dynamique de touches semi-transparentes. Ces dernières sont ajoutées autour de la touche qui vient d'être pressée. Les lettres sont déterminées en fonction de ce qui a été saisi précédemment grâce à un système de prédiction. Les résultats a priori de ce système laisse espérer un gain d'au moins 30% de la vitesse de saisi.

MOTS CLES : clavier logiciel, système de prédiction, transparence

ABSTRACT

This article presents a system of optimization for soft keyboard. It rests on the dynamic addition of semi-transparent keys. These one are added around the key witch has just been typed. The letters are given according to keyboarding letters previously and thanks to a predictive system. Results a priori of this system let hope a profit from at least 30% of the text entry speed.

KEYWORDS: soft keyboard, transparency, predictive system

INTRODUCTION

A l'origine, les claviers logiciels ont été créés pour permettre aux personnes handicapées moteur d'utiliser un ordinateur. En effet grâce à ces claviers intégrés au système, ces personnes peuvent via un dispositif de pointage saisir du texte. Avec l'émergence des dispositifs mobiles dépourvus de clavier physique (par exemple, les assistants personnels), les claviers logiciels sont maintenant aussi utilisé sur ce type de système à l'aide d'un stylet.

Cependant les performance d'un clavier physique de type AZERTY sont très nettement supérieures à celles de son homologue logiciel. Ceci peut être expliqué par le fait que l'on peut saisir du texte avec plusieurs doigts sur un clavier physique ce qui limite les déplacements entre chaque touche. Limitée au seul dispositif de pointage, la saisie avec un clavier logiciel voit ses performances diminuer considérablement à cause du temps de déplacement d'une touche à l'autre.

Ainsi pour palier ces faibles performances, des recherches sont menées pour essayer d'améliorer la vitesse de saisie de texte sur clavier logiciel. La principale problématique consiste à ne pas ajouter de charge cognitive trop forte à l'utilisation du clavier logiciel, afin de ne perturber l'utilisateur.

Dans cette optique, nous proposons un système basé sur des touches ajoutées en fonction du contexte afin d'améliorer la vitesse de saisie de texte. Ces touches sont semi-transparentes et de couleur vive pour attirer l'œil de l'utilisateur sans lui faire perdre ses repères par rapport au clavier initial.

Au travers de cet article, nous montrerons en quoi consiste les recherches actuelles en optimisation de claviers logiciels. Puis nous expliquerons comment évaluer ces systèmes. Enfin nous decrirons notre système KeyGlasses, les premiers résultats théoriques de celui-ci et les travaux à venir.

TECHNIQUES D'OPTIMISATION

Optimiser la disposition des touches

Un des axes de recherche consiste à essayer d'optimiser la disposition des touches. Ainsi en placant côte à côte les touches qui ont le plus de chance de se succéder, on diminue les distances parcourues lors de la saisie et par conséquent le temps de cette dernière.

Plusieurs claviers ont été réalisés dans cette optique. Certains ont été conçus en utilisant les tables bi-grammes donnant les probabilités pour qu'une lettre succède à une

autre. On peut notamment citer le clavier commercialisé Fitaly¹ ou encore le clavier OPTI [4] (cf. fig. 2). D'autres claviers mettent en jeu en plus des modèles linguistiques des techniques normalement utilisées en physique telles que l'algorithme Metropolis qui a permis de concevoir le clavier de même nom [9].



Fig. 1 : Clavier OPTI [2]

Une autre façon de limiter les déplacements consiste à mettre plusieurs lettres sur la même touche comme par exemple [2]. Ces claviers sont dits ambigus. Bien que pratiques pour les appareils mobiles du fait de leur faible encombrement, ces claviers se trouvent moins efficaces pour la saisie de texte pas des personnes handicapées [8].

Mais cette optimisation de la disposition des touches, bien que meilleure en moyenne, peut être inappropriée pour un ensemble de mots, du fait de son statisme durant la saisie.

Des modèles dynamiques

Des systèmes plus dynamiques permettent d'améliorer la vitesse de saisie de texte. Le système Sybilette [6] permet de réorganiser l'ensemble des touches en fonction de ce qui vient d'être saisi. Mais ce clavier logiciel ne fonctionne pas avec un dispositif de pointage. Il est basé sur un défilement automatique linéaire des touches et l'utilisateur appuie sur un bouton poussoir lorsque le système est sur la bonne touche. Grâce à ce défilement automatique, l'utilisateur a le temps de percevoir les changements du clavier. Mais ce genre de système ne donnerait pas de bon résultats avec un système de pointage car la charge cognitive supplémentaire pour « apprendre » le nouveau clavier serait trop importante.

Dans notre cas d'utilisation (i.e. avec un système de pointage), le dynamisme des claviers est plus souvent basé sur l'ajout dynamique de menus au cours de la saisie. Ces menus peuvent être constitués soit par la liste des mots les plus probables comme dans [5], soit par une proposition de lettres par exemple sous forme de pie-menu [3].

Cependant la présentation de ces menus peut poser des problèmes à l'utilisateur. En effet, l'opacité du menu

cache une partie du clavier et son apparition peut gêner l'utilisateur dans sa saisie.

MODELES D'EVALUTION

Dès les années 50, Fitts [1] détermina une équation permettant d'estimer a priori le temps pour se déplacer d'une cible C_i à une autre C_j (cf. Fig. 1). Le temps de déplacement entre ces deux touches est de :

$$MT = a + b \log_2(D_{ij} / W_j + 1) \quad (1)$$

Où D_{ij} représente la distance entre les deux cibles, W_j la taille de la cible C_j et a et b sont des coefficients fixés de manière empirique.

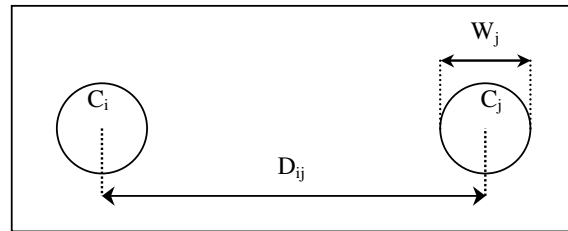


Fig. 2 : Distance D_{ij} entre une cible C_i et une cible C_j

Cette loi de Fitt a bien sûr été utilisée pour évaluer la vitesse de saisie de texte des claviers logiciels. Mackenzie, un des précurseurs dans le domaine a estimé les coefficients à $a=0.127$ et $b = 1 / 4.9$ [2].

Plus récemment, Zhai a proposer une formule permettant d'évaluer l'ensemble du clavier [9] en prenant en compte la distance entre chaque touche, pondérée par la probabilité qu'elles se succèdent. Cette formule permet ainsi de comparer aisément les différents claviers.

Cependant cette loi ne prend en compte que le temps mis pour aller d'une touche à l'autre. Ceci implique que l'utilisateur maîtrise le clavier qu'il utilise et qu'il ne passe pas du temps à chercher les caractères sur le clavier.

NOTRE SYSTEME : LES KEYGLASSES

Idée de départ

De la même façon que les pie-menus, nous souhaitons apporter près de la touche saisie, des touches supplémentaires qui pourraient accélérer la saisie. Mais à la différence de [3] qui propose toujours le même pie-menu, nos touches supplémentaires sont dépendantes de ce qui vient d'être saisi.

En outre, afin de ne pas trop encombrer l'écran et la charge de l'utilisateur, nous souhaitons limiter le nombre de touches supplémentaires. Dans un premier temps, seules les 4 caractères les plus probables sont affichés.

¹ <http://www.fitaly.com>

Le principe

Au fur et à mesure que l'utilisateur va saisir du texte, notre système lui propose autour de la touche qu'il vient de presser les quatre lettres qu'il a le plus de chance de saisir juste après celle-ci (cf fig. 3). Ces lettres sont déterminées grâce à un système de prédiction basé sur ce qui vient d'être saisi. Ce sont ces lettres rajoutées en fonction du contexte que nous appelons KeyGlasses du fait de leur semi-transparence.

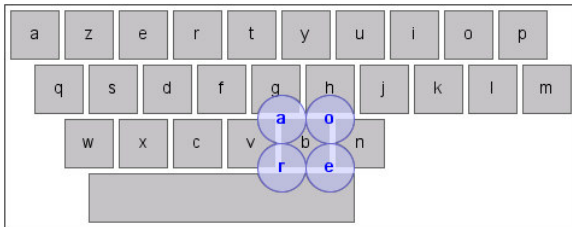


Fig. 3 : Disposition des touches après avoir saisi la lettre 'b'

La semi-transparence des KeyGlasses, permet de toujours voir l'intégralité du clavier. Leur couleur plus vive que le clavier permet d'attirer l'attention de l'utilisateur vers ces touches. Si une des KeyGlass lui convient il peut l'actionner de la même manière que les touches fixes : par un simple clic. Dans le cas où l'utilisateur souhaiterait accéder à la touche se trouvant sous une KeyGlass, il peut y parvenir par un double clic.

Ce système peut être récurrent : les KeyGlasses apparaissent après toute saisie de caractère, que ce soit à partir d'une touche fixe du clavier ou d'une KeyGlass (cf. fig. 4). Dans le cas d'une KeyGlass, cette dernière reste en arrière plan des 4 nouvelles KeyGlasses proposées.

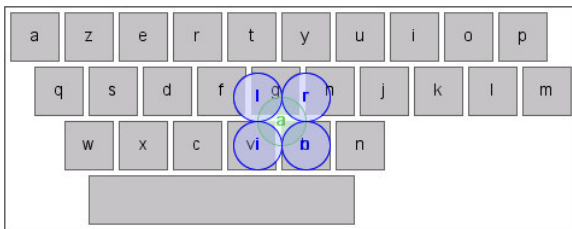


Fig. 4 : Cas de récurrence

Fonctionnement et architecture du système

Lorsqu'une touche est pressée, la position du clic est fournie au système qui détermine la position des KeyGlasses suivantes. En parallèle, la lettre saisie est envoyée au système de prédiction qui détermine les 4 lettres les plus probables et les renvoie au système d'affichage des KeyGlasses. Celui-ci attribue chacune d'elles à une KeyGlass et affiche le tout.

Ce système est indépendant de l'organisation des touches du clavier. La disposition des touches ainsi que les

caractéristiques de celles-ci sont décrites dans un fichier au format XML. Ainsi, il est possible de modifier la disposition des touches du clavier très simplement.

De plus, le système d'affichage des KeyGlasses est indépendant du système de prédiction. Il est donc possible de changer ce dernier pour, par exemple, adapter très rapidement ce système à une autre langue.

RESULTATS A PRIORI

A l'heure actuelle, nous n'avons pas mené d'expérimentations poussées avec des utilisateurs. Afin de tester rapidement l'efficacité de notre système, nous avons simulé la saisie d'un ensemble de mots sur celui-ci. Pour permettre la comparaison, nous avons réalisé la simulation sur un clavier logiciel de type AZERTY et ce même clavier avec notre système KeyGlasses. Pour ce dernier, nous avons simulé deux fois cette saisie en essayant deux systèmes de prédiction différents : un sous forme d'arbre lexicographique et l'autre sous forme de bigramme.

La simulation a été faite sur un peu plus de 30 000 mots différents pris dans des articles du Monde et de l'encyclopédie Universalis. Pour chacun de ces mots, nous avons calculé la distance qu'aurait normalement dû parcourir l'utilisateur via son dispositif de pointage pour passer d'une touche à l'autre. Cette distance a été calculée en supposant que l'utilisateur appuie systématiquement au centre de la touche. A chaque fois que la lettre à saisir faisait partie des KeyGlasses, c'est cette dernière qui a été sélectionnée.

Le tableau 1 présente les résultats des simulations réalisées sur les 3 systèmes. En plus de la distance moyenne (exprimée en pixels), nous apportons une estimation de la vitesse moyenne (en pixels/seconde) nécessaire pour saisir ces mots. Celle-ci a été calculée en utilisant la fonction de MacKenzie présentée précédemment.

	AZERTY	KeyGlass	
		arbre lexico	Bi-gramme
Distance	1184	453	589
Vitesse	3.22	2.14	2.34

Tab. 1 : Performance des différents systèmes

Le tableau 2 présente les gains (en pourcentage) pour la distance et la vitesse par rapport au clavier AZERTY. La dernière ligne donne le taux d'utilisation des KeyGlasses, si celle-ci ont été utilisées à chaque fois que cela est possible.

	Arbre Lexicographique	Bi-gramme
Distance	61.74	50.25
Vitesse	33.54	27.33
Utilisation	68,03	56.35

Tab. 2 : Performance des systèmes de prédiction

Les résultats que nous apportons dans le tableau 1, sont les résultats d'une utilisation optimale de ce système. Ils laissent envisager de bonnes performances. Cependant ils ne prennent pas en compte ni le fait que l'utilisateur ne prendra pas forcément la KeyGlass à chaque fois, ni le temps de recherche visuelle d'une touche.

Nous pouvons observer dans le tableau 2 l'importance du système de prédiction. En effet, même si les deux systèmes de prédiction donnent de bons résultats, ceux-ci varient de manière assez importante.

TRAVAUX FUTURS

Afin de confirmer ces premiers résultats et d'évaluer la surcharge cognitive engendrée par le système, nous allons maintenant tester notre système auprès d'utilisateurs finaux. Nous viserons plus particulièrement deux types de populations : d'une part les personnes handicapées moteur et d'autre part les utilisateurs de dispositifs mobiles.

En plus de ces expérimentations, il nous reste à valider certains points de notre système. Par exemple, il serait intéressant de déterminer le nombre optimal de KeyGlasses affichées. Les différents systèmes de prédiction seront aussi à tester pour voir quel est celui qui sera le plus performant.

Nous étendrons aussi nos tests sur différents types de claviers, pour déterminer si le gain s'atténue avec un clavier avec une disposition des touches optimisée.

CONCLUSION

Notre système qui peut être considéré comme un complément pour un clavier logiciel, apporte au vu des premiers résultats a priori un gain significatif sur la distance et donc sur la vitesse de saisie de texte.

BIBLIOGRAPHIE

1. Fitts P.M., The information capacity of the human motor system in controlling the amplitude of the movement, *Journal of experimental psychology* 47, pp.381-391, 1954.
2. Green N, Kruger J., Faldu C., St. Amant R., A reduced QWERTY keyboard for mobile text entry, *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '04*, pp. 1429-1432, Vienne, 2004.
3. Isokoski P. Performance of menu-augmented soft keyboards, *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '04*, pp. 423-430, Vienne, 2004.
4. MacKenzie, I. S., & Zhang, S. Z. The design and evaluation of a high performance soft keyboard. *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '99*, pp. 25-31. New York: ACM, 1999.
5. Masui T., An efficient text input method for pen-based computers, *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '98*, pp. 328-335, Los Angeles, 1998
6. Schadle I., Le Pevedic B., Antoine J.-Y., Poirier F., Sybillette : système de prédiction de lettre pour l'aide à la saisie de texte, *TALN'2001, atelier thématique "Handicap et Ingénierie Linguistique*, Tours, France, vol 2., pp. 233-242, 2001.
7. Soukoreff W., MacKenzie I.S., Theoretical upper and lower bounds on typing speeds using a stylus and keyboard, *Behavior & Information Technology*, 14, 370-379
8. Vigouroux N., Vella V., Truillet Ph., Raynal M., Evaluation of AAC for text input by two populations: normal versus handicapped motor persons (8th ERCIM UI4All, Vienna (Österreich), 28-29 June 2004
9. Zhai, S., M. Hunter, and B.A. Smith. The Metropolis Keyboard - an exploration of quantitative techniques for virtual keyboard design. *Proceedings of The 13th Annual ACM Symposium on User Interface Software and Technology (UIST)*. San Diego, California: ACM. pp. 119-218, 2000.