

---

# Le système KEYGLASS

## Système d'ajout dynamique de touches sur clavier logiciel

**Mathieu Raynal\***

\* *Équipe IHCS - Laboratoire IRIT*  
*Université Paul Sabatier, 31062 Toulouse Cedex 9*  
*raynal@irit.fr*

---

*RÉSUMÉ. Cet article présente le système KeyGlass : un système de saisie de texte avec ajout dynamique de caractères en fonction de ceux précédemment saisis. Le système de prédiction que nous utilisons pour optimiser notre système est basé sur l'utilisation conjointe d'un arbre lexicographique et d'un système utilisant des bigrammes.*

*Nous présentons dans cet article les différentes étapes qui nous ont conduit vers ce système de prédiction. Enfin nous étudions, au travers de deux évaluations (l'une théorique et l'autre avec des utilisateurs), l'utilité et l'efficacité de notre système lors d'une tâche de saisie de texte. Les résultats montrent une réduction importante de la distance parcourue par le pointeur sur le clavier logiciel. En revanche, les utilisateurs sont un peu moins rapides pour saisir du texte.*

*ABSTRACT. This paper presents the KeyGlass system: a text entry system with dynamic addition of characters based on those previously entered. The prediction system that we use to optimize our system is based on the joint use of a lexicographic tree and a system using bigrams.*

*We present in this article the different steps that led us to this prediction system. Finally we study, through two experiments (one theoretical and the other one with users), the usefulness and effectiveness of our system during a task of text copy. The results show a significant reduction in the distance covered by the pointer on the soft keyboard. However, users are slower to enter text.*

*MOTS-CLÉS : claviers logiciels, prédiction de caractères, évaluation.*

*KEYWORDS: soft keyboard, character prediction, evaluation.*

---

## 1. Introduction

La saisie de texte est une activité essentielle lors de l'utilisation d'un ordinateur. Pour les personnes ayant un fort handicap des membres supérieurs, cette activité a été rendue possible grâce aux claviers dits « logiciels » qui sont des applications logicielles permettant de saisir des données. Les plus connus et utilisés sont les représentations numériques de claviers physiques affichés à l'écran. Pour saisir du texte avec ce type de clavier, deux actions sont nécessaires : se déplacer vers le caractère à saisir, puis le sélectionner.

Le déplacement du pointeur, pour pointer le caractère à saisir, peut se faire à l'aide de tout type de dispositif de pointage : souris, trackball, touchpad, mais aussi des dispositifs plus « lourds » comme ceux de suivi du regard (Hansen *et al.*, 2004 ; Majaranta et Riih , 2002) ou les mouvements de la t te (LoPresti *et al.*, 2003 ; LoPresti *et al.*, 2000).

Actuellement le clavier logiciel le plus fr quemment utilis  a une disposition de touches et de caract res qui est identique   celle des claviers physiques Azerty ou Qwerty (selon la culture). Si cette disposition pr sente l'avantage d' tre utilis e depuis plus d'un si cle et donc d' tre connue du plus grand nombre, elle a un inconv nient majeur qui r s de dans l' loignement des caract res les plus fr quemment utilis s. Ce probl me ne se posait pas sur les claviers physiques avec l'utilisation de deux doigts diff rents pour saisir successivement deux caract res  loign s. Cependant, l'utilisation d'un seul pointeur pour s lectionner les caract res impose de d placer ce dernier entre chaque caract re   saisir. L' loignement des caract res les plus fr quemment utilis s entra ne une augmentation de la distance   parcourir avec le pointeur, ce qui engendre une diminution de la vitesse de saisie pour l'utilisateur, mais aussi des fatigues motrices voire oculaires (B rard, 2004 ; Vella *et al.*, 2005).

Pour minimiser les distances   parcourir lors de la saisie sur claviers logiciels, deux axes principaux ont  t   tudi s :

- agencer diff remment les caract res sur le clavier de mani re   rapprocher les caract res qui se succ dent le plus r guli rement. De nouveaux claviers ont ainsi  t  g n r s soit de mani re intuitive (par exemple, les claviers FITALY et OPTI (MacKenzie et Zhang, 1999)), soit   partir d'algorithmes d'optimisation (comme les claviers METROPOLIS (Zhai *et al.*, 2000) et GAG (Raynal et Vigouroux, 2005a)) ;

- l'autre technique d'optimisation consiste   utiliser un syst me de pr diction de mots. Une des utilisations les plus classiques d'un syst me de pr diction consiste   proposer un ensemble de mots les plus probables par rapport au d but de la saisie de l'utilisateur. Dans le cas o  un des mots propos s est celui recherch  par l'utilisateur, celui-ci peut alors valider ce mot au lieu de finir de le saisir via les touches du clavier. Ceci a pour but de diminuer le nombre de caract res   saisir. Plusieurs syst mes ont  t  propos s sur ce principe. On peut notamment citer les syst mes KeyStrokes<sup>1</sup> (B rard, 2004 ; Niemeijer, 2005), HandiAS (Le P v dic, 1997 ; Maurel *et al.*, 2001), Reactive

1. <http://www.assistiveware.com/keystrokes.php>

Keyboard (Darragh *et al.*, 1990), SybiMot (Schadle, 2003) ou FASTY (Beck *et al.*, 2004).

Ces deux méthodes présentent des avantages, mais aussi des inconvénients : dans le premier cas, quelle que soit la disposition des caractères, certains seront toujours éloignés sur le clavier. Pour la seconde proposition, la position de la liste soumettant les mots n'est pas toujours optimale. Celle-ci est soit placée sur un des côtés du clavier, comme par exemple pour KeyStrokes, soit elle recouvre une partie du clavier comme proposé par Masui dans son système POBox (Masui, 1998).

Dans le but de diminuer au maximum les déplacements à effectuer avec le dispositif de pointage, nous proposons un système à mi-chemin entre ces deux grands axes de recherche que nous venons de voir : le système KeyGlass qui propose les caractères les plus probables à proximité du dernier saisi.

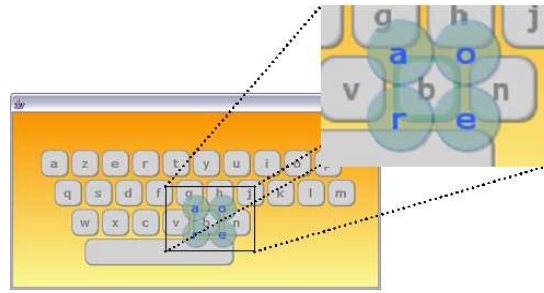
Après avoir décrit notre système et le système de prédiction que nous avons utilisé, nous présenterons les résultats de deux expérimentations : la première est une évaluation théorique ; la seconde est une expérimentation menée avec 6 personnes handicapées des membres supérieurs. Nous analyserons ces résultats, puis les discuterons avant de proposer différents axes de développement et de recherche pour nos travaux futurs.

## 2. Le système KeyGlass

### 2.1. *Le principe*

Nous avons conçu un système qui propose à l'utilisateur des caractères supplémentaires au fur et à mesure de la saisie associés à des touches ajoutées dynamiquement à proximité du dernier caractère saisi. Nous nommerons ce système KeyGlass. Nous avons fait le choix de proposer après chaque caractère saisi les caractères qui ont le plus de probabilité de lui succéder. Même si le temps de recherche visuelle du caractère peut être plus long, nous souhaitons proposer le plus souvent possible le bon caractère à proximité du dernier saisi, et ainsi réduire au maximum les distances de saisie. Pour augmenter les probabilités de diminuer les distances, le système fonctionne aussi par récurrence : après chaque nouvelle saisie d'un caractère, que ce soit sur une touche fixe ou une touche ajoutée dynamiquement (appelée KeyGlass), de nouveaux caractères sont proposés autour de ce dernier.

À l'instar des *toolglasses* (Bier *et al.*, 1993) qui proposent une palette semi-transparente d'outils à proximité du pointeur, nous avons choisi d'afficher les caractères supplémentaires sur des touches rondes semi-transparentes (cf figure 1). Nous avons préféré retenir la semi-transparence pour donner à l'utilisateur une vision globale du clavier et ainsi ne jamais masquer un caractère par l'affichage d'un autre qui viendrait se positionner par dessus. La sélection des caractères se fait de la même manière (par un clic sur ladite touche) que ce soit un caractère associé à une touche du clavier fixe ou à une touche KeyGlass.



**Figure 1.** *Quatre KeyGlasses positionnées après la saisie du caractère 'b'*

## 2.2. Architecture du système de saisie

### 2.2.1. Un système modulaire

Nous avons conçu le système de manière modulaire pour pouvoir changer une partie du fonctionnement sans avoir à changer le reste. Notre système est constitué de trois parties (voir figure 2) :

- **la partie fixe du clavier logiciel** : elle concerne l'ensemble des touches qui ne bougeront pas pendant la saisie. Les caractères associés à ces touches sont aussi fixes. Cet ensemble de touches/caractères est décrit dans un format XML. Nous avons retenu un langage XML car il permet de modifier facilement les caractéristiques du clavier de base. Nous pouvons ainsi changer la forme, la position, les couleurs d'une touche ainsi que le caractère qui y est associé. Cette partie fixe est par conséquent complètement indépendante des KeyGlasses qui peuvent être ajoutées par dessus ;

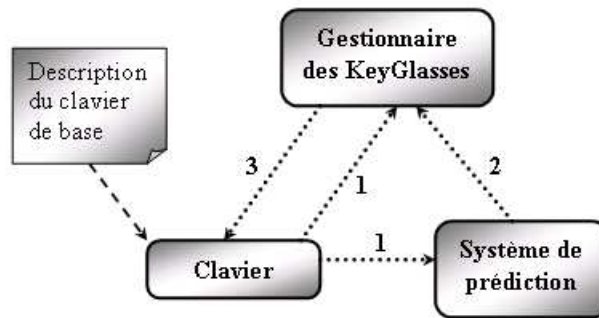
- **le système de prédiction** : il reçoit les caractères saisis par l'utilisateur. À partir de ces derniers, il renvoie au gestionnaire d'affichage des KeyGlasses un classement des caractères qui ont le plus de probabilité de succéder à ce qui vient d'être saisi ;

- **le gestionnaire d'affichage des KeyGlasses** : il gère l'affichage des KeyGlasses au dessus du clavier fixe. Il a à sa charge : la position des KeyGlasses, leur forme, leur couleur. Il leur associe le caractère en fonction de ce qu'il reçoit du système de prédiction.

### 2.2.2. Mode de fonctionnement

La figure 2 montre le principe de fonctionnement du système KeyGlass. Les chiffres sur les flèches donnent l'ordre de transmission des messages entre les différents modules.

1) Lorsqu'un caractère est saisi sur le clavier logiciel, celui-ci est envoyé au système de prédiction. Dans le même temps, les coordonnées du pointeur au moment de



**Figure 2.** Architecture du système KeyGlass

la sélection du caractère<sup>2</sup> sont envoyées au gestionnaire de KeyGlasses.

2) Le système de prédiction calcule la probabilité de chaque caractère de succéder au dernier caractère reçu, puis les classe par ordre croissant en fonction de leur probabilité d'apparaître. Il renvoie alors ce classement au gestionnaire d'affichage de KeyGlasses.

3) Enfin, le gestionnaire d'affichage des KeyGlasses, après avoir déterminé la position des futures KeyGlasses, affecte à chacune d'elles un caractère et renvoie toutes ces informations au clavier logiciel pour que celui-ci les affiche.

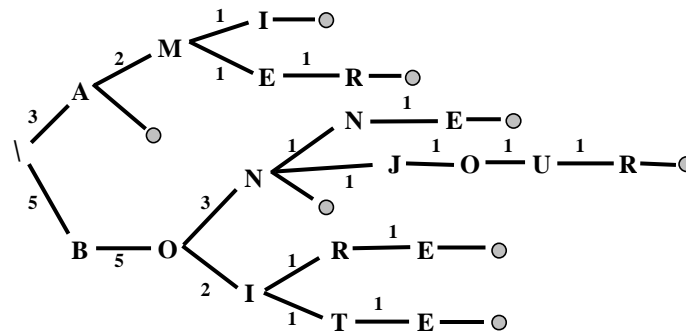
Notons que les KeyGlasses une fois affichées au dessus du clavier sont considérées par le système comme appartenant au clavier logiciel. Ainsi, comme nous l'avons mentionné dans la section 2.1, ce processus peut fonctionner de manière récursive. L'appui sur une KeyGlass pour saisir un caractère entraîne exactement le même processus que celui que nous venons de décrire ci-dessus.

### 2.2.3. Protocole de communication inter-module

Nous avons utilisé le bus logiciel IVY (Buisson *et al.*, 2002) pour la communication entre les différents modules. Pour faciliter le remplacement d'un module par un autre, nous avons défini un protocole de communication entre les modules. Ainsi, chaque module connaît exactement ce qu'il peut recevoir et/ou envoyer. Le remplacement d'un module par un autre est facile : il suffit que ce dernier soit connecté sur le bus IVY et respecte le protocole de communication.

Afin d'être compatible avec la plate-forme E-ASSISTE (Raynal *et al.*, 2005), ce protocole de communication reprend les règles de celui de E-ASSISTE. Ainsi les di-

<sup>2</sup> Cette sélection peut être caractérisée, par exemple, par la pression d'un bouton du dispositif de pointage.



**Figure 3.** *Arbre lexicographique construit à partir des mots a, ami, amer, bon, bonne, bonjour, boire, boîte*

vers messages respectant ce protocole peuvent être aussi collectés par la plate-forme E-ASSISTE et permettre une analyse *a posteriori* de la saisie de l'utilisateur et du fonctionnement du système KeyGlass.

### 3. Le système de prédiction utilisé

#### 3.1. Description des systèmes testés

Nous avons choisi d'étudier plus particulièrement deux systèmes de prédiction.

– **Un système à base de bigrammes**, qui propose pour un caractère donné, les caractères qui ont le plus de chance de lui succéder. Ce procédé ne prend en compte pour la prédiction que le caractère qui vient d'être saisi. On retrouve ainsi toujours, pour un caractère donné, les mêmes caractères prédits. Ces caractères prédits ont l'avantage de pouvoir être appris par une personne qui les utiliserait régulièrement. Notre table de bigrammes a été déduite de la base de données BDLEX (De Calmès et Pérennou, 1998).

– **Un système à base d'arbre lexicographique**, construit à partir d'un ensemble de mots (cf. figure 3). Chaque mot est représenté par un chemin de la racine à une feuille de l'arbre, où chaque caractère est un nœud de l'arbre. Deux mots ayant le même préfixe utilisent le même chemin initial pour éviter de créer trop de nouveaux nœuds. Pour les mots qui sont les préfixes d'autres mots, une feuille spéciale 'fin de mot' (cercle grisé sur la Figure 3) a été créée. Comme (Ménier et Poirier, 2001), nous avons choisi de donner un poids à chaque arête de l'arbre. À chaque mot ajouté, les arêtes entre les différents nœuds traversés sont incrémentées de 1. Ceci permet de donner une importance à chaque nœud. Notre arbre lexicographique a été construit à partir des 140 000 mots les plus fréquents de la langue française.

Ce système prend en compte le début du mot qui vient d'être saisi en parcourant l'arbre en fonction de ce qui a déjà été saisi. Par rapport à ce début de mot, il classe les

caractères qui peuvent lui succéder en fonction de leur probabilité d'apparition (plus la valeur de l'arête est grande, plus le caractère a de chance d'apparaître). L'avantage de ce système est qu'il ne propose, après le début de mot déjà saisi, que les caractères qui sont susceptibles de former un mot connu. L'inconvénient est que si l'arbre lexicographique ne contient pas un mot dans son corpus d'apprentissage, ou si l'utilisateur fait une faute lors de la saisie de son mot sans la corriger, la prédiction risque d'être fautive par rapport à ce que veut saisir l'utilisateur.

### 3.2. Évaluation de ces systèmes

Afin de déterminer le système de prédiction le plus performant à intégrer dans notre système KeyGlass, nous avons fait une simulation par ordinateur pour évaluer les performances de notre système avec chacun d'eux. Celle-ci consistait à simuler la saisie d'un ensemble de mots. Nous avons ainsi réalisé trois sessions de simulation : une avec un clavier AZERTY (limité aux 26 caractères de l'alphabet et l'espace), une avec le système KeyGlass et le système de prédiction à base de bigrammes, et une autre avec le système KeyGlass et le système de prédiction à base d'arbres lexicographiques.

Chaque session de simulation a consisté à saisir l'ensemble des mots de la version 3 de l'Officiel du Scrabble (ODS)<sup>3</sup>. Ce dictionnaire comporte 364 370 mots, soit 4 005 510 caractères<sup>4</sup>. Nous avons simulé la saisie avec l'utilisation de quatre KeyGlasses. Pour déterminer la pertinence des deux systèmes de prédiction, nous avons étudié plus particulièrement le taux de caractères correctement prédits : c'est-à-dire le nombre de caractères qui peuvent être saisis directement à partir d'une KeyGlass.

### 3.3. Analyse des résultats

Le tableau 1 présente les résultats de ces simulations où :

- *nb caractères* représente le nombre de caractères correctement prédits par le système et affichés sur une KeyGlass, parmi les 4 005 510 à saisir ;
- *tx prédiction* donne les performances du système de prédiction en taux de bonne prédiction.

Comme nous pouvons le constater sur le tableau 1, les performances *a priori* seraient meilleures avec un système de prédiction basé sur un arbre lexicographique. Cependant, une étude plus minutieuse de ces résultats montre que ces deux systèmes sont assez complémentaires. Pour cela, nous avons réalisé un décompte des caractères correctement prédits après chaque caractère saisi dans le mot et ce, pour chaque mot. Cette étude montre une progression différente pour les deux systèmes. L'évo-

3. <http://www.monunivers.com/scrabble/ods.htm>

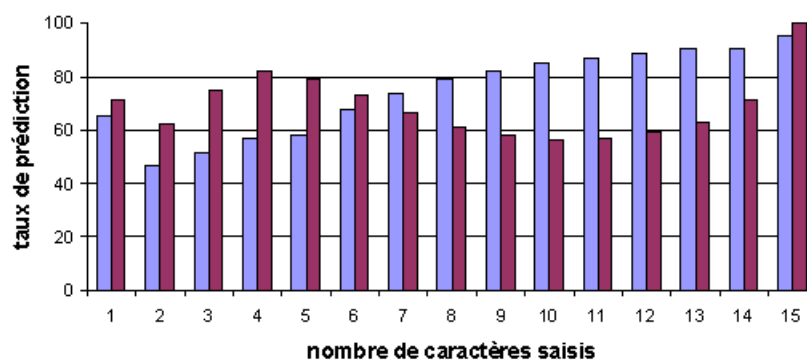
4. Le nombre de caractères a été calculé en prenant pour chaque mot la taille de la chaîne de caractères plus un caractère espace à la fin.

Clavier	<i>Nb caractères</i>	<i>Tx prédiction</i>
Bigramme	<b>2 449 235</b>	61,15
Arbre lexico	<b>2 499 692</b>	62,40

**Tableau 1.** *Résultat des simulations avec différents systèmes de prédiction*

lution des performances des deux systèmes est présentée sous forme d'histogramme sur la figure 4. L'axe des abscisses représente le nombre de caractères du mot qui ont été saisis au moment de la prédiction. L'axe des ordonnées donne le pourcentage de caractères correctement prédits par rapport au nombre de caractères à saisir. Nous voyons ainsi qu'un système de prédiction de type arbre lexicographique a de meilleures performances que celui basé sur les bigrammes sur les six ou sept premières lettres. En revanche, pour les mots longs, le système de prédiction à base de bigrammes est meilleur sur les dernières lettres du mot que l'arbre lexicographique. Ceci peut s'expliquer par le fait que l'arbre lexicographique a été construit à partir des 140 000 mots les plus fréquents de la langue française<sup>5</sup> qui ne comportent pas l'ensemble des formes fléchies que l'on retrouve dans l'Officiel du Scrabble. Les désinences des formes fléchies étant souvent les mêmes, on obtient une fréquence élevée

5. Corpus issu d'un ensemble d'articles du journal *Le Monde*, et de l'*Encyclopédie Universalis*



**Figure 4.** *Taux de prédiction correcte en fonction du système de prédiction et de l'avancée dans le mot (en bleu le système de prédiction à base de bigrammes, en bordeaux l'arbre lexicographique)*



de ces bigrammes. Ainsi, le système de prédiction bigramme prend en compte ces fréquences élevées de bigrammes.

Les avantages respectifs de ces deux systèmes nous ont amené à évaluer le gain que nous apporterait une combinaison des deux.

### 3.4. Le système de prédiction final

À partir de cette analyse, nous avons modélisé un système de prédiction prenant en compte à la fois l'arbre lexicographique et le bigramme. Pour chaque prédiction de caractères, le système prend les résultats en sortie de ces deux systèmes et les pondère respectivement par un coefficient  $P_a$  pour l'arbre lexicographique et  $P_b$  pour le bigramme. Ces deux valeurs sont comprises entre 0 et 1 et  $P_a = 1 - P_b$ .

Ainsi pour connaître les meilleures valeurs à attribuer à  $P_a$  et  $P_b$ , nous avons réalisé une série de simulations où nous faisons varier le poids des deux systèmes de prédiction. Nous avons commencé avec  $P_a = 0,95$  et  $P_b = 0,05$  puis nous les avons respectivement décrémenté et incrémenté par pas de 0,05 jusqu'à obtenir les valeurs inverses du départ ( $P_a = 0,05$  et  $P_b = 0,95$ ).

$P_a$	$P_b$	P	$P_a$	$P_b$	P
0,95	0,05	74,83 %	0,45	0,55	72,90 %
0,90	0,10	74,86 %	0,40	0,60	72,52 %
0,85	0,15	74,69 %	0,35	0,65	72,07 %
0,80	0,20	74,53 %	0,30	0,70	71,65 %
0,75	0,25	74,36 %	0,25	0,75	71,01 %
0,70	0,30	74,23 %	0,20	0,80	70,42 %
0,65	0,35	73,94 %	0,15	0,85	69,58 %
0,60	0,40	73,73 %	0,10	0,90	68,48 %
0,55	0,45	73,52 %	0,05	0,95	65,78 %
0,50	0,50	73,23 %			

**Tableau 2.** Performance du système de prédiction en fonction de la pondération de l'arbre lexicographique et du bigramme

Le tableau 2 présente les résultats obtenus pour l'ensemble des pondérations testées. La meilleure solution consiste à prendre en compte les prédictions de l'arbre lexicographique à 90 % et celle du bigramme à 10 %. Cette solution a été retenue pour être le système de prédiction de base du système KeyGlass.

### **3.5. Le nombre de caractères proposés**

Afin de savoir quel nombre de KeyGlasses il est intéressant d'afficher après chaque caractère, nous avons réalisé une dernière simulation. Celle-ci consistait aussi à simuler la saisie de l'ensemble des mots de l'Officiel du Scrabble. Nous avons réalisé douze sessions : nous sommes partis d'une simulation avec une seule KeyGlass (un seul caractère prédit), et nous en avons rajouté une de plus par session, jusqu'à en avoir douze.

Nous avons pris pour cette simulation le système de prédiction optimal précédemment calculé (c'est-à-dire le système couplant arbre lexicographique et bigramme en les pondérant respectivement par 0,9 et 0,1). À titre de comparaison, nous avons également réalisé la même série de simulations en proposant toujours les mêmes caractères (nous avons pris les caractères les plus fréquemment utilisés pour la langue française).

Le tableau 3 présente l'ensemble de ces résultats. Nous avons choisi pour la suite de notre conception de retenir quatre KeyGlasses. Ce choix s'explique par le fait que nous obtenons un taux d'utilisation des KeyGlasses de 74,76 % avec quatre caractères prédits, ce qui est à peu près équivalent au taux que l'on obtiendrait avec dix caractères fixes, soit plus que ce qui est proposé par (Isokoski, 2004). De plus, ceci nous paraît être un bon compromis entre le taux de prédiction et l'affichage de celles-ci à l'écran (cela permet d'en mettre une à chaque coin de touche, par exemple).

Pour la conception de notre système de saisie, nous avons fait le choix d'afficher quatre KeyGlasses et de proposer les caractères qui y sont associés à partir d'un système de prédiction hybride entre un arbre lexicographique et une table bigramme.

Nous allons maintenant étudier et évaluer par des expérimentations l'efficacité et l'utilisabilité de ces quatre KeyGlasses.

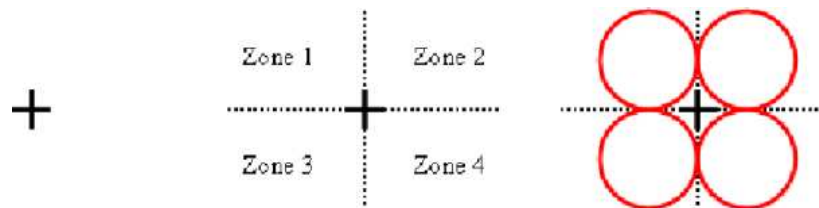
## **4. Première évaluation du système**

### **4.1. Description du principe d'affichage des KeyGlasses**

Dans la première version du clavier optimisé par le système KeyGlass, nous avons cherché à minimiser les distances à parcourir. Le gestionnaire de KeyGlasses positionne les quatre KeyGlasses autour du pointeur du dispositif de pointage. La position des KeyGlasses est calculée en fonction des coordonnées du pointeur au moment de la pression sur une touche normale ou une KeyGlass du clavier (voir figure 5). À partir de ces coordonnées, quatre zones sont formées pour positionner les KeyGlasses (fi-

Nb	Prédiction		Caractères fixes	
1	1 422 439	35,51 %	500 675	12,50 %
2	2 296 198	57,33 %	844 699	21,09 %
3	2 704 554	67,52 %	1 209 069	30,19 %
4	2 994 645	74,76 %	1 534 980	38,32 %
5	3 195 508	79,78 %	1 864 918	46,56 %
6	3 299 010	82,36 %	2 138 823	53,40 %
7	3 384 816	84,50 %	2 397 716	59,86 %
8	3 441 196	85,91 %	2 629 438	65,65 %
9	3 484 235	86,99 %	2 840 360	70,91 %
10	3 522 484	87,94 %	2 979 008	74,37 %
11	3 555 267	88,76 %	3 111 639	77,68 %
12	3 574 288	89,23 %	3 199 115	79,87 %

**Tableau 3.** Performance du système de prédiction en fonction du nombre de KeyGlasses disponibles



**Figure 5.** Désignation des positions des KeyGlasses

gure 5). Les Keyglasses sont placées de manière à coller aux axes. Cette manière de positionner les KeyGlasses a l'avantage d'être au plus près du pointeur sans qu'il n'y ait de recouvrement entre les KeyGlasses.

Pour cette première itération, nous avons considéré le clavier AZERTY comprenant les 26 caractères de l'alphabet latin ainsi que le caractère espace. Le système de prédiction est celui retenu suite aux différentes simulations de la section 3. Il ne repose que sur la prédiction de ces 27 caractères. La figure 1 montre un exemple du clavier utilisé pour cette première itération ainsi que les KeyGlasses qui ont été positionnées après la saisie du caractère 'b'.

#### 4.2. *Évaluation théorique*

Cette évaluation théorique a pour objectif de calculer le gain que pourrait apporter l'utilisation du système à un utilisateur. Rappelons les deux hypothèses faites pour ce système : d'une part, diminuer les distances à parcourir avec le dispositif de pointage, et d'autre part, augmenter la vitesse de saisie de texte.

Ce sont ces deux variables (distance et vitesse) que nous avons cherché à quantifier par cette simulation. À des fins de comparaison, nous avons réalisé une simulation en utilisant le clavier de base sans utilisation de KeyGlasses, puis la même simulation avec les KeyGlasses.

Pour ce système, l'évaluation théorique conduite pour l'évaluation du clavier GAG (Raynal et Vigouroux, 2005a), basée sur la formule de Soukoreff (Soukoreff et MacKenzie, 1995), n'est pas possible. En effet, l'aspect dynamique du système ne permet pas de prévoir toutes les situations possibles comme sur un clavier fixe. La saisie de chaque mot est unique : nous avons réalisé l'évaluation en simulant la saisie de l'ensemble des mots de l'Officiel du Scrabble.

Pour calculer les distances de chaque mot, nous avons pris à chaque fois comme coordonnées de pression d'une touche le centre de la touche concernée. Ce choix est justifié par l'étude de Zhai et ses collègues (Zhai *et al.*, 2002) qui a montré que lors de la sélection d'une touche, la répartition des impacts d'un stylet sur celle-ci était concentrée autour du centre de la touche. La distance d'un mot est alors obtenue en additionnant la longueur de chaque segment séparant les centres de deux touches qui viennent d'être pressées successivement. Cette simulation de saisie de texte est dite parfaite :

- 1) la saisie de chaque mot est effectuée sans erreur de frappe ;
- 2) dans le cas de la simulation avec le système KeyGlass, si le caractère à saisir est prédit et apparaît sur une KeyGlass, la simulation choisit à chaque fois le caractère se trouvant sur la KeyGlass. Sinon il le prend sur la disposition de touches du clavier de base.

Pour le calcul du temps, on distingue généralement deux classes d'utilisateurs : les novices et les experts (Soukoreff et MacKenzie, 1995). Les performances obtenues par ces deux types d'individus sont généralement considérées comme les bornes des performances que l'on peut espérer avoir avec le système en question. Notre système étant composé de deux parties (un clavier fixe et des touches supplémentaires dépla-

<i>Type de clavier</i>		<i>AZERTY</i>	<i>KeyGlass</i>	<i>Différence</i>
<i>Distance en pixels</i>		<b>1 206,19</b>	<b>483,15</b>	- 59,94 %
<i>Temps</i>	Novice	<b>10 776 ms</b>	<b>10 802 ms</b>	0,24 %
	Expert/Novice	<b>3 270 ms</b>	<b>5 493 ms</b>	67,98 %
	Expert	<b>3 270 ms</b>	<b>1 860 ms</b>	- 43,12 %

**Tableau 4.** Résultat en distance et temps des simulations selon les classes d'utilisateur

cées dynamiquement au cours de la saisie), nous avons choisi d'étudier trois cas :

1) l'utilisateur est novice à la fois avec le clavier fixe et les KeyGlasses. En plus du temps de déplacement, nous comptons pour chaque caractère à saisir, un temps de recherche visuelle de ce dernier parmi les 31 possibilités (les 27 caractères du clavier fixe plus les quatre KeyGlasses) ;

2) l'utilisateur est expert sur le clavier de base, mais novice avec les KeyGlasses : pour ce type d'utilisateur, le temps de recherche visuelle se limite à la recherche parmi les quatre KeyGlasses (la recherche visuelle est considérée comme nulle sur le clavier fixe, vu que l'on considère l'utilisateur comme expert sur ce support) ;

3) l'utilisateur est expert avec l'ensemble du système : le temps de recherche est considéré comme négligeable. Le temps pour saisir un mot se limite au temps de déplacement d'un caractère à un autre.

L'estimation du temps pour se déplacer est prédite par la loi de Fitts (Fitts, 1954) alors que celle du temps de recherche visuelle d'une touche est formulée par la loi de Hick-Hyman (Hick, 1952 ; Hyman, 1953). Nous avons choisi de prendre comme valeur pour ces différentes lois :

- $a = 0$ ,  $b = 4,5$ , et  $MT_{repeat} = 0,127$  pour la loi de Fitts. Nous avons repris les mêmes valeurs calculées pour le pointage de cible avec une souris pour la loi de Fitts.
- $a = 0$  et  $b = 0,2$  pour la loi de Hick-Hyman ; ce sont les valeurs utilisées pour la loi de Hick-Hyman pour prévoir le temps maximal qu'un utilisateur va passer à rechercher un caractère.

Ainsi, le temps nécessaire pour saisir un mot est la somme des temps de déplacement et de recherche visuelle pour chaque caractère saisi.

Le tableau 4 présente les résultats de ces simulations. La colonne AZERTY donne les résultats pour le clavier fixe sans utilisation de KeyGlasses, la colonne *KeyGlass* représente ceux de ce même clavier avec utilisation du système KeyGlass. La colonne *Différence* donne en pourcentage la différence qui existe entre une utilisation du clavier fixe sans et avec KeyGlasses. Enfin, les distances sont exprimées en nombre de

pixels par mot, et le temps représente le temps moyen en millisecondes pour saisir un mot.

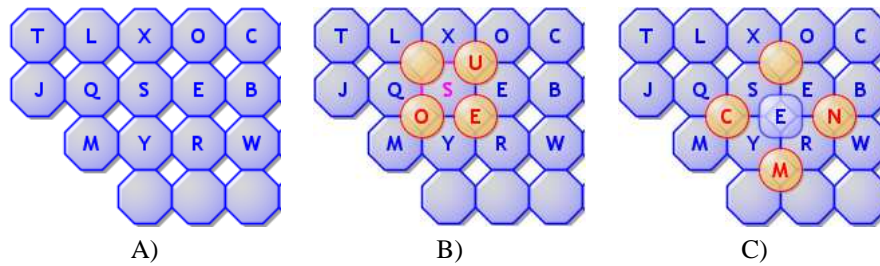
Rappelons que le taux de prédiction de notre système de prédiction est de l'ordre de 75 % dans cette configuration. Nous constatons que si l'utilisateur saisit, dès que cela est possible, les caractères positionnés sur les KeyGlasses, il peut réduire la distance à parcourir d'environ 60 %. En revanche, les gains en vitesse de saisie sont moins importants. Si un utilisateur connaît la disposition touches/caractères du clavier sur lequel il va utiliser pour la première fois les KeyGlasses, le gain en temps est négatif (- 67,98 %). Par contre, si l'utilisateur ne connaît pas la disposition des caractères du clavier sur lequel il va saisir, les temps de saisie seront alors quasi identiques entre le clavier avec ou sans le système KeyGlass (en revanche, l'utilisateur y gagnera en distance). À partir du moment où l'utilisateur est en situation d'expert (c'est-à-dire après plusieurs heures d'utilisation) avec le clavier et le système KeyGlass, l'apport de ce dernier peut alors devenir intéressant même en terme de vitesse : on peut ainsi espérer une diminution du temps de l'ordre de 40 % dans le meilleur des cas.

## **5. Deuxième évaluation : expérimentation avec des personnes handicapées moteur**

Une première expérimentation avait été réalisée avec en majorité des sujets ne comportant aucun problème moteur. Cette première expérimentation avait comme but de faire ressortir les problèmes potentiels d'utilisation. Les résultats de cette expérimentation ont été publiés dans (Raynal et Vigouroux, 2005b). À partir de ces résultats et des retours utilisateurs nous avons donc choisi de réaliser une seconde version de notre prototype présenté sur la figure 6. La principale particularité réside dans la forme des touches du clavier de base qui sont octogonales. Cette caractéristique donne l'avantage, lorsque les touches sont collées les unes aux autres, de laisser des espaces vides à la place des quatre coins traditionnels d'une touche rectangulaire. Nous avons utilisé ces espaces comme seuls endroits autorisés pour l'affichage d'une KeyGlass. Les KeyGlasses sont affichées dans les quatre espaces les plus proches de la touche qui vient d'être sélectionnée. L'espace laissé libre par les touches octogonales évite le problème de lisibilité des caractères à la fois du clavier mais aussi des KeyGlasses.

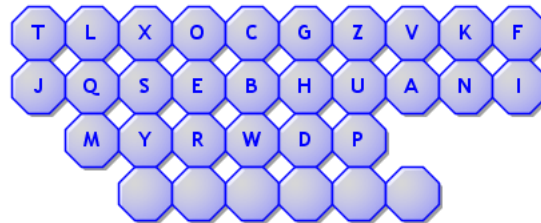
### **5.1. Protocole expérimental**

Nous cherchons à comparer au travers de cette expérimentation les caractéristiques d'un clavier logiciel seul et celles du même clavier couplé au système KeyGlass. L'hypothèse principale que nous formulons est que notre système KeyGlass permet de diminuer de manière significative la distance à parcourir avec le pointeur pour saisir du texte. En second lieu, nous regarderons si le temps de saisie diminue au cours du temps, s'il existe une différence sur l'effet d'apprentissage et le temps de recherche visuelle d'une touche respectivement sur un clavier logiciel classique et avec notre système KeyGlass.



**Figure 6.** Clavier pour la version 2 des KeyGlasses : A. clavier simple ; B. KeyGlasses qui apparaissent après la saisie du 's' ; C. KeyGlasses qui apparaissent après la saisie du 'e' qui était placé sur une KeyGlass.

Sachant que les sujets sont novices avec le système KeyGlass, nous souhaitons qu'ils le soient aussi avec le clavier logiciel seul. C'est pourquoi nous avons choisi de ne pas prendre le clavier logiciel Azerty classique dont la disposition des caractères est connue de tous. Pour le remplacer, nous avons créé un clavier logiciel identique à l'Azerty sauf par sa disposition des caractères (nous avons gardé la barre espace en bas du clavier car différente des autres par sa forme). Pour assurer des performances en vitesse de saisie à peu près identiques, nous avons utilisé le modèle de Soukoreff et Mackenzie (Soukoreff et MacKenzie, 1995) pour évaluer la disposition des caractères. Ainsi, nous avons généré aléatoirement le positionnement des touches et ce jusqu'à obtenir une vitesse de saisie théorique équivalente à celle de la disposition Azerty à  $10^{-3}$  près. Pour éviter toute ambiguïté avec le clavier logiciel Azerty classique, nous appellerons ce clavier le clavier SEB<sup>6</sup> (cf. figure 7).



**Figure 7.** Clavier SEB : clavier de base utilisé pour l'expérimentation

Nous avons testé notre système sur deux profils de sujets : d'une part, des personnes valides, et d'autre part des personnes ayant un handicap des membres supérieurs. Chaque population de sujets était constituée de six sujets. Nous ne discuterons

6. du fait que ces trois lettres se suivent sur la seconde ligne

dans cet article que des résultats des personnes handicapées. L'étude complète est détaillée dans (Raynal, 2005).

Chaque sujet avait vingt sessions à réaliser. Chaque session consistait à effectuer un exercice de saisie une fois avec le clavier SEB seul et une fois (le même exercice) avec le clavier SEB couplé au système KeyGlass. Pour éviter certains effets d'apprentissage, l'ordre d'utilisation des claviers était alterné. Ainsi, une fois sur deux le sujet commençait par le clavier seul et la fois suivante il commençait par le clavier couplé au système KeyGlass. La moitié de chaque groupe de sujets a commencé les exercices en utilisant les claviers dans un certain ordre, et l'autre moitié dans l'autre ordre possible. Enfin, un exercice d'entraînement a été proposé avant le premier exercice de la première session. L'exercice de saisie consistait à recopier trente mots que nous avons déterminés à l'avance parmi une liste de mots apparaissant les plus fréquemment dans la littérature enfantine. Ainsi les mots à saisir ne devaient pas poser de difficultés majeures (pouvant entraîner des hésitations) lors de la recopie par les sujets.

Chaque sujet a réalisé l'ensemble de l'expérimentation à domicile. Les sujets étaient libres de réaliser une session de l'expérimentation quand ils le souhaitaient. Les seules contraintes qui leur ont été données étaient de laisser au moins 3 heures entre deux sessions d'expérimentation, et les sessions devaient être espacées d'au plus 48 heures.

#### 5.1.1. *Le système*

L'ensemble du clavier et du système KeyGlass a été réalisé en Java. Le recueil des données a été réalisé avec la plate-forme E-Assiste. Pour chaque session d'expérimentation, l'ensemble des données recueillies était enregistré dans un fichier en respectant un formalisme XML.

#### 5.1.2. *Le matériel*

Toutes les expérimentations ayant lieu chez les sujets, nous n'avons pu contrôler le matériel utilisé. Le dispositif de pointage utilisé a été le même pour chaque sujet valide. Par contre, les sujets handicapés n'ayant pas tous la même motricité, les dispositifs de pointage sont assez variés : deux sujets avec une souris classique, deux avec une trackball, un avec un joystick et le dernier un touchpad.

### 5.2. *Analyse des résultats*

Voici la présentation des résultats obtenus à partir des données recueillies lors des vingt sessions des six sujets. Les résultats statistiques présentés dans cette section ont été obtenus par des tests paramétriques d'analyse de la variance (test ANOVA). Ces traitements ont été réalisés avec le logiciel SYSTAT 11.

#### 5.2.1. *Comparaison des claviers testés*

Le tableau 5 présente les résultats généraux de cette expérimentation. Ces derniers



Variables	SEB	SEB + KeyGlass
Taux de prédiction	-	69,11
Taux d'utilisation	-	96,02
Distance	1 106	511
Durée	6 895	8 282
Erreur	1,015	1,024

**Tableau 5.** Résultats généraux de l'expérimentation de la seconde itération

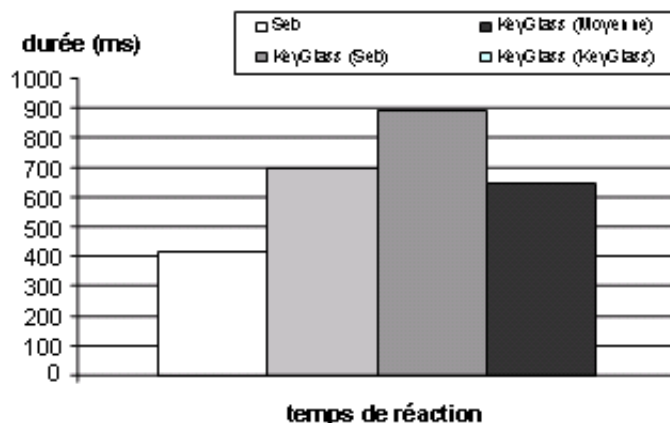
représentent la moyenne sur l'ensemble des sessions de tous les sujets. Les différentes lignes du tableau correspondent à :

- la *Prédiction* qui donne le pourcentage de caractères prédits correctement par notre système ;
- l'*Utilisation* qui donne sous forme de pourcentage, le taux d'utilisation des Key-Glasses par les utilisateurs. Ce taux est calculé à partir du nombre de fois où le caractère à saisir se trouve sur une des KeyGlasses par rapport au nombre de fois où l'utilisateur l'a utilisé ;
- la *Distance* qui est la distance moyenne (donnée en pixels) parcourue pour saisir un mot ;
- la *Durée* exprimée en millisecondes, traduit la durée moyenne dont a eu besoin l'utilisateur pour saisir un mot de l'expérimentation ;
- l'*Erreur* qui donne le KSPC<sup>7</sup> (MacKenzie, 2002), c'est-à-dire le nombre de touches frappées divisé par le nombre de caractères correctement saisis.

Il est important de mentionner que les utilisateurs ont fortement utilisé les Key-Glasses dès lors que cela était possible (plus de 97 % d'utilisation). Ceci se traduit par une diminution des distances très significative ( $p < 0,001$ ) (plus de 55 % de diminution entre le clavier SEB seul et ce même clavier avec le système KeyGlass). En revanche, la durée nécessaire pour saisir un mot reste encore plus élevée avec l'utilisation de notre système KeyGlass (il y a une augmentation d'environ 13 % par rapport au clavier SEB seul). La différence de vitesse de saisie entre les deux claviers est aussi significative ( $p < 0,001$ ). Enfin, le taux d'erreur reste raisonnable pour les deux claviers avec environ 3 % d'erreurs pour le clavier SEB avec le système KeyGlass.

D'après la loi de Fitts (Fitts, 1954), si l'on diminue la distance qui nous sépare de la cible à pointer, le temps pour atteindre cette dernière devrait diminuer. Or, comme

7. Keystrokes per Character



**Figure 8.** Temps de réaction moyen pour saisir un caractère

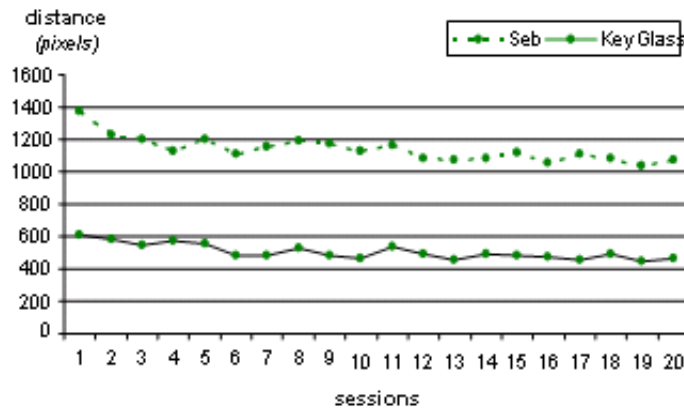
nous venons de le voir, la distance diminue en moyenne de près de 55 %, alors que la durée pour saisir un mot augmente en moyenne de 13 %. Deux interprétations peuvent être avancées pour expliquer ce phénomène :

- d’une part, les erreurs, qui peuvent être un frein pour la vitesse de saisie de texte lorsque celles-ci sont trop nombreuses. Le tableau 5 montre que le nombre d’erreurs est un peu plus élevé lors de l’utilisation du système KeyGlass, mais reste cependant faible (3 % d’erreurs lors de saisie de texte est un taux d’erreur normal),

- d’autre part, le temps de recherche d’un caractère peut être un facteur important de perte de temps lors d’une tâche de saisie de texte, surtout lorsque les utilisateurs sont novices avec le système de saisie de texte. Pour estimer ce temps de recherche visuelle, nous avons calculé le temps de réaction de l’utilisateur. Nous définissons ce dernier comme étant le temps qui sépare le moment où l’utilisateur a saisi un caractère et le moment suivant cette saisie où il amorce un mouvement avec le dispositif de pointage. Ainsi, nous considérons qu’à partir du moment où l’utilisateur commence à déplacer le pointeur, c’est qu’il a trouvé le caractère qu’il veut atteindre et qu’il commence le mouvement de pointage de celui-ci.

La figure 8 présente les temps de réaction moyens. Il est intéressant de noter que ce temps de réaction est plus long lors des tâches de saisie sur le clavier SEB associé au système KeyGlass (bâton gris clair) que sur le clavier SEB seul (bâton blanc) avec respectivement 700 ms et 415 ms pour réagir (soit une augmentation de 68,7 %). On peut interpréter ce temps supplémentaire comme étant celui nécessaire pour visualiser les quatre caractères qui se trouvent sur les KeyGlasses.

Enfin, nous avons aussi voulu savoir, pour le clavier SEB avec le système KeyGlass, s’il y avait une différence de temps de réaction entre un mouvement qui abouti-



**Figure 9.** Évolution de la distance parcourue pour saisir un mot au cours des sessions

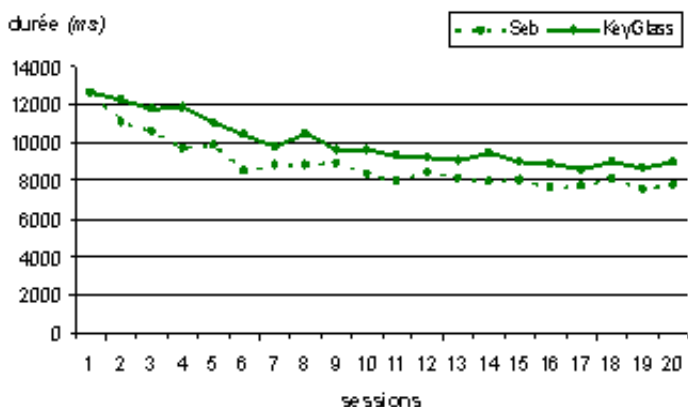
rait à la saisie d'un caractère sur KeyGlass et un à une touche fixe du clavier SEB. Ces deux temps de réaction sont représentés respectivement sur la figure 8 par les bâtons noir et gris foncé. On peut ainsi constater que les sujets mettent plus de temps pour amorcer un mouvement quand il s'agit d'aller sur une touche fixe. Une hypothèse est qu'ils commencent par vérifier sur les quatre KeyGlasses que le caractère qu'ils cherchent ne s'y trouve pas avant de rechercher sur les touches fixes.

Une des explications possibles de la perte de temps peut être le temps de recherche qui semble plus long lors de l'utilisation des KeyGlasses. Cependant, ce temps de recherche devrait normalement diminuer au cours de sessions d'utilisation : l'utilisateur connaissant normalement de mieux en mieux la position des caractères. Les résultats que nous venons de présenter sont une moyenne sur l'ensemble des sessions. C'est pourquoi nous allons maintenant regarder l'évolution des différentes variables au cours des sessions pour voir s'il existe un effet d'apprentissage.

### 5.2.2. Effet des sessions d'apprentissage

Les figures 9, 10 et 11 présentent respectivement l'évolution de la distance, de la durée moyenne de saisie d'un mot et des temps de réaction au cours des vingt sessions.

La courbe représentant la distance moyenne à parcourir pour saisir un mot sur le clavier SEB est en pointillé, celle concernant ce même clavier avec le système KeyGlass est en traits continus. Nous pouvons voir que la distance sur le clavier SEB diminue de manière assez conséquente (- 21 %) lors des six premières sessions. Nous pouvons imaginer que cette diminution de distance sur ce clavier, qui pourtant reste fixe, est en partie due à la culture des sujets qui ont l'habitude d'utiliser un clavier



**Figure 10.** Évolution au cours des sessions de la durée moyenne pour saisir un mot

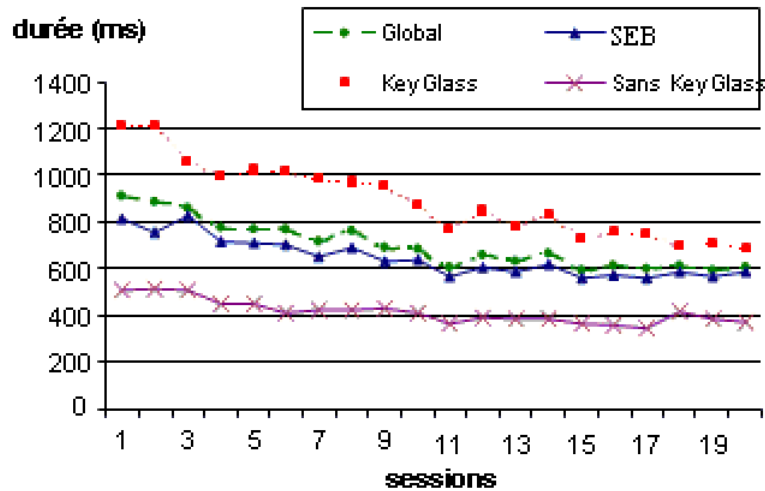
Azerty et qui anticipaient leur mouvement vers des caractères en pensant les trouver à la même place qu’habituellement sur le clavier Azerty.

La figure 10 présente l’évolution de la durée nécessaire pour saisir un mot. La diminution importante de la distance que nous avons constatée sur les six premières sessions pour le clavier SEB seul se traduit aussi par une diminution du temps. Cette diminution est même plus importante au niveau du temps : elle représente un gain sur la durée de 33 % entre les sessions 1 et 6. Dans le même intervalle, le gain de temps sur ce clavier avec le système KeyGlass n’est que de 20 %. Une fois ces six ou sept premières sessions passées, le gain jusqu’à la vingtième session est alors moins important, avec un gain de 12 % sur le clavier SEB avec ou sans le système KeyGlass.

Nous pouvons penser que l’essentiel de l’apprentissage de la position des caractères sur un clavier se fait pendant les dix premières sessions. Il apparaît aussi qu’il est plus facile d’apprendre une disposition de caractères fixes qu’une disposition de caractères dynamiques qui varie en fonction de la saisie. Cependant, cette seconde étant plus compliquée à apprendre, on pourrait penser que l’apprentissage continuerait pendant plus longtemps que pour une disposition de caractères fixes. Or, au regard des courbes de la figure 10, il apparaît que pour les deux claviers, la courbe d’apprentissage est la même.

Pour avoir une vue plus globale du comportement des sujets, la figure 11 présente les temps de réaction au cours des vingt sessions pour l’ensemble des expériences réalisées.

Nous présentons les temps de réaction pour le clavier SEB seul (courbe avec croix) et pour le clavier SEB associé au système KeyGlass. Nous pouvons voir que le temps



**Figure 11.** Évolution au cours des sessions du temps de réaction

de réaction sur le clavier SEB seul n'a pas de diminution aussi significative lors des six ou sept premières sessions que celle que l'on a pu constater pour la distance et le temps. Ceci renforce l'hypothèse que nous avons formulée au moment de l'interprétation des distances, en évoquant une possible anticipation en pensant connaître le clavier : les sujets pensant au clavier Azerty anticiperaient un mouvement, et le modifieraient en cours de déplacement. Ceci expliquerait que les temps de réaction soient assez faibles dès la première session et que la distance soit plus élevée lors des six ou sept premières sessions.

En revanche, on peut constater que le temps de réaction est bien plus élevé au départ avec les KeyGlasses, et que la diminution du temps de réaction sur les six ou sept premières sessions est plus prononcée (avec près de 20 % de temps en moins après sept sessions). Il apparaît donc que l'affichage des KeyGlasses force l'utilisateur à regarder les caractères présents sur celles-ci et supprime l'effet d'anticipation après la saisie d'un caractère.

Enfin, les courbes avec des triangles et des carrés présentent la décomposition des temps de réaction lors de l'utilisation du système KeyGlass avec le clavier SEB. Elles présentent l'évolution des temps de réaction qui aboutissent à la saisie d'un caractère respectivement sur une touche fixe du clavier SEB et sur une KeyGlass. On peut ainsi constater que lors des premières sessions, c'est le temps de réaction pour aller saisir un caractère sur une KeyGlass qui est le plus pénalisant (environ 1 200 ms contre 800 ms pour le temps de réaction avant la saisie sur une touche fixe). La diminution de cette durée est par ailleurs plus importante pour la saisie sur KeyGlass que pour la saisie sur touches fixes. On peut supposer que cette diminution est due en partie

à l'effet d'apprentissage : le sujet commence à savoir au bout de quelques sessions s'il va trouver (ou non) sur les KeyGlasses le caractère qu'il veut saisir. S'il sait que le caractère n'y est pas, il va alors directement pouvoir chercher sur le clavier fixe le caractère voulu.

## 6. Conclusion

Au regard des résultats et après discussion avec les principaux intéressés (les personnes handicapées), le système KeyGlass apporte bien une forte diminution des distances qui est appréciable pour ces personnes. Notre première hypothèse qui était de diminuer les distances est largement atteinte.

En revanche, malgré cette diminution des distances, le temps mis par les sujets pour saisir un mot demeure plus long lorsqu'ils utilisent le système KeyGlass. L'hypothèse que nous faisons concernant la diminution du temps au cours des sessions n'est que partiellement vérifiée. En effet, le temps de saisie diminue bien lors des premières sessions, mais l'apprentissage n'a pas plus d'effet que pour un clavier logiciel classique.

Après discussion informelle avec les sujets, il apparaît que si le principe de rajouter des touches à proximité est grandement apprécié, le fait que les caractères changent en fonction de la saisie du mot les perturbe énormément dans leur vitesse de saisie de texte. Le fait de proposer sur ces touches les caractères les plus probables ne semble pas être la meilleure stratégie : ni en utilisabilité, ni en gain en terme de vitesse de saisie de texte.

## 7. Travaux futurs

Nous envisageons maintenant de faire de nouveaux tests en modifiant cette fois-ci la proposition des caractères associés aux KeyGlasses. L'objectif est de permettre à l'utilisateur d'apprendre les caractères qui se trouvent sur les KeyGlasses pour qu'il puisse anticiper son mouvement. Nous envisageons de tester deux nouvelles stratégies de proposition de caractères.

– Proposer, quel que soit le caractère saisi, toujours les mêmes caractères sur les KeyGlasses, comme c'est le cas, par exemple, sur le marking-menu proposé par Isokoski (Isokoski, 2004). Cette méthode diminuera le nombre de fois où il est possible d'utiliser les KeyGlasses, mais c'est la méthode la plus simple pour apprendre ces caractères supplémentaires.

– L'autre solution est un intermédiaire entre ce que nous avons déjà réalisé et la solution d'Isokoski. Elle consiste à proposer pour un caractère donné toujours les mêmes caractères prédits. Ainsi pour chaque caractère, il y aura une configuration de caractères différente. Cela fait un peu plus de configurations à apprendre, mais il paraît possible de les apprendre.

Ainsi nous testerons ces deux possibilités pour voir quel est le meilleur compromis entre la réduction de la distance, le temps gagné et le temps d'apprentissage.

Une fois que nous aurons décidé de la stratégie à adopter pour les caractères associés sur les KeyGlasses, il sera intéressant de tester notre système KeyGlass sur un clavier logiciel plus complet : la limitation que nous avons faite aux 26 caractères de l'alphabet latin et à l'espace n'est pas représentative de la saisie de texte pour la langue française. C'est pourquoi la dernière étape de notre processus itératif sera l'évaluation de notre système sur un clavier complet.

## 8. Bibliographie

- Beck C., Seisenbacher G., Edelmayer G., Zagler W., « First User Test Results with the Predictive Typing System FASTY », in K. Miesenberger, J. Klaus, W. Zagler, D. Burger (eds), *9th ICCHP : Computer Helping People with Special Needs*, Lecture Notes in Computer Science, LCNS 3118, p. 813-819, 2004.
- Bier E., Stone M., Pier K., Buxton W., DeRose T., « Toolglass and magic lenses : the see-through interface », *SIGGRAPH '93 : Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, p. 73-80, 1993.
- Buisson M., Bustico A., Chatty S., Colin F., Jestin Y., Maury S., Mertz C., Truillet P., « Ivy : un bus logiciel au service du développement de prototypes de systèmes interactifs », *14<sup>e</sup> Conférence Francophone sur l'Interaction Homme-Machine (IHM'02)*, ACM Press, New York, NY, USA, p. 223-226, 2002.
- Bérard C., « Clavier-écran : concevoir avec les utilisateurs », *Handicap 2004*, p. 83-88, 2004.
- Darragh J., Witten I., James M., « The Reactive Keyboard : A Predictive Typing Aid », *Computer*, vol. 23, n° 11, p. 41-49, 1990.
- De Calmès M., Pérennou G., « BDLEX : a Lexicon for Spoken and Written French », *1st International Conference on Language Resources & Evaluation*, Grenade, A. Rubio, N. Gallardo, R. Castro, A. Tejada, ELRA, Paris, p. 1129-1136, 28-30 mai, 1998.
- Fitts P., « The information capacity of the human motor system in controlling the amplitude of movement », *Journal of Experimental Psychology*, vol. 47, p. 381-391, 1954.
- Hansen J., Torning K., Johansen A., Itoh K., Aoki A., « Gaze typing compared with input by head and hand », *ETRA'2004 : Proceedings of the Eye tracking research & applications symposium on Eye tracking research & applications*, ACM Press, New York, NY, USA, p. 131-138, 2004.
- Hick W., « On the rate of gain of information », *Quarterly Journal of Experimental Psychology*, vol. 4, p. 11-36, 1952.
- Hyman R., « Stimulus information as a determinant of reaction time », *Journal of Experimental Psychology*, vol. 45, p. 188-196, 1953.
- Isokoski P., « Performance of menu-augmented soft keyboards », *CHI '04 : Proceedings of the 2004 conference on Human factors in computing systems*, ACM Press, p. 423-430, 2004.
- Le Pévédic B., Prédiction morphosyntaxique évolutive dans un système d'aide à la saisie de textes pour des personnes handicapées physique, PhD thesis, Ecole doctorale Sciences pour l'ingénieur de Nantes, 1997.

- LoPresti E., Brienza D., Angelo J., Gilbertson L., « Neck range of motion and use of computer head controls », *Journal of Rehabilitation Research and Development*, vol. 40, n° 3, p. 199-212, mai/juin, 2003.
- LoPresti E., Brienza D., Angelo J., Gilbertson L., Sakai J., « Neck range of motion and use of computer head controls », *Assets '00 : Proceedings of the fourth international ACM conference on Assistive technologies*, ACM Press, New York, NY, USA, p. 121-128, 2000.
- MacKenzie I., « KSPC (Keystrokes per Character) as a Characteristic of Text Entry Techniques », *Mobile HCI '02 : Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction*, Springer-Verlag, p. 195-210, 2002.
- MacKenzie I., Zhang S., « The design and evaluation of a high-performance soft keyboard », *CHI '99 : Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, p. 25-31, 1999.
- Majaranta P., Riih  K., « Twenty years of eye typing : systems and design issues », *ETRA '02 : Proceedings of the symposium on Eye tracking research & applications*, ACM Press, New York, NY, USA, p. 15-22, 2002.
- Masui T., « An efficient text input method for pen-based computers », *CHI '98 : Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, p. 328-335, 1998.
- Maurel D., Rossi N., Thibault R., « HandiAS : un syst me multilingue pour l'aide   la communication des personnes handicap es », *TALN'01*, vol. 2, p. 203-212, 2001.
- M nier G., Poirier F., « Syst me adaptatif de pr diction de texte », *TALN 01*, Tours, p. 213-222, 2-5 juillet, 2001.
- Niemeijer D., « In memoriam of Christian B rard : Striving for effort reduction through on-screen keyboard word prediction », *Workshop AAATE*, 2005.
- Raynal M., M thodes d'optimisation et  valuation pour une meilleure utilisabilit  des claviers logiciels, Th se de doctorat, Universit  Paul Sabatier, Toulouse, France, d cembre, 2005.
- Raynal M., Maubert S., Vigouroux N., Vella F., Magnien L., « E-Assiste : A platform Allowing Evaluation of Text Input System », *3rd Int. Conf. on Universal Access in Human-Computer Interaction (UAHCI 2005)*, Las Vegas, USA, 22/07/05-27/07/05, Lawrence Erlbaum Associates (LEA), ISBN 0-8058-5807-5, juillet, 2005. Pages de la publication : CDRom.
- Raynal M., Vigouroux N., « Genetic algorithm to generate optimized soft keyboard », *CHI '05 : CHI '05 extended abstracts on Human factors in computing systems*, ACM Press, New York, NY, USA, p. 1729-1732, 2005a.
- Raynal M., Vigouroux N., « KeyGlasses : Semi-transparent keys to optimize text input on virtual keyboard », in A. Pruski, H. Knops (eds), *Assistive technology : from virtuality to reality - 8th European conference for the advancement of assistive technology in europe (AAATE 2005)*, Lille, France, 06/09/05-09/09/05, IOS Press, ISBN 1-58603-543-6, p. 713-717, septembre, 2005b.
- Schadle I., SIBYLLE : syst me linguistique d'aide   la communication pour les personnes handicap es, PhD thesis, Universit  de Bretagne Sud, 2003.
- Soukoreff R., MacKenzie I., « Theoretical Upper and Lower Bounds on Typing Speed Using a Stylus and Soft Keyboard », *Behavior and Information Technology*, vol. 14, n° 6, p. 370-379, 1995.
- Vella F., Vigouroux N., Truillet P., « SOKEYTO : a design and simulation environment of software keyboards », in A. Pruski, H. Knops (eds), *Assistive technology : from virtuality to*



*reality - 8th European conference for the advancement of assistive technology in europe (AAATE 2005)*, Lille, France, IOS Press, ISBN 1-58603-543-6, p. 723-727, 6-9 septembre, 2005.

Zhai S., Hunter M., Smith B., « The metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design », *UIST '00 : Proceedings of the 13th annual ACM symposium on User interface software and technology*, ACM Press, p. 119-128, 2000.

Zhai S., Sue A., Accot J., « Movement model, hits distribution and learning in virtual keyboarding », *CHI '02 : Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, p. 17-24, 2002.