

État de l'art des claviers physiques et logiciels pour la saisie de texte

State of art of physical and software keyboards for text entry

Benoît MARTIN (1), Isabelle PECCI (2)

(1) LITA, Université Paul Verlaine-Metz, France
benoit.martin@univ-metz.fr

(2) LITA, Université Paul Verlaine-Metz, France
pecci@univ-metz.fr

Résumé. L'attrait croissant pour l'informatique mobile tend à multiplier les modes de saisie, mais ce n'est pas la seule motivation. Les besoins des utilisateurs changent, ils s'élargissent, car les utilisateurs eux-mêmes ont changé. Des besoins très spécifiques à un contexte d'utilisation ou à un handicap rendent les claviers standard inutilisables. Cet article présente un état de l'art des dispositifs offrant une alternative aux claviers standard. Il propose un classement des différentes solutions rencontrées, parmi les claviers physiques ou logiciels. Plusieurs critères sont pris en compte pour analyser ces différentes approches : la présentation avec le type de saisie, la prédiction éventuelle, le contexte d'utilisation et les performances. Cet article a pour but de montrer une large panoplie des claviers afin d'avoir un aperçu sur différentes approches et pouvoir se faire une idée sur les possibilités offertes par chacune d'elles.

Mots-clés. Clavier physique, clavier logiciel, taxinomie, technique d'interaction.

Abstract. The pull for mobile computing encourages the amount of methods for text entry. It is not the unique motivation. The needs for users have changed ; they grow because the users have changed. Now, very specific needs to a context of use or to an impairment make standard keyboard unusable. This article presents a state of the art of the keyboard-replacement solutions. It aims at classifying solutions encountered, among physical keyboards, and among software keyboards. Several benchmarks are considered to study these various keyboards: look, kind of text entry, prediction, context of use and performance. This article shows a wide range of keyboards. The goal is to have a look at many different approaches aims to understand the various keyboards solutions and aims to help to choose best keyboards according context of use and users.

Keywords. Physical keyboard, software keyboard, taxonomy, interaction technique.

1 Introduction

De nos jours, le clavier est un élément encore souvent indispensable quand on doit interagir avec un système informatique. Les claviers Qwerty et Azerty de nos ordinateurs, très répandus, ne constituent cependant pas la panacée pour toutes les utilisations. En effet, à cause de la place qu'ils occupent, de la disposition de leurs touches ou bien pour des raisons de sécurité et/ou de maintenance, de nombreux systèmes informatiques ne sont plus équipés de claviers. Parmi ces systèmes on trouve notamment les bornes interactives (pour éviter la maintenance) ainsi que les téléphones portables et les assistants numériques ; même si des modèles récents possèdent un clavier complet ils restent encore minoritaires. L'essor de ces systèmes et de ces nouvelles technologies montre le besoin de disposer de méthodes alternatives de saisie de texte pour se substituer à un clavier standard.

Il existe déjà d'autres méthodes de saisie, dont se servent certaines catégories d'utilisateurs (les personnes handicapées, ou encore les utilisateurs d'assistants numériques, ...). Notre travail vise à proposer un état de l'art des claviers pour saisir du texte. Il s'agit d'une extension de l'étude proposée par (Poirier et Schadle, 2005) qui s'intéressait à la saisie de texte dans un contexte précis : les interfaces réduites. Le présent état de l'art couvre tout type de claviers qu'ils soient physiques, logiciels, dédiés ou non à un contexte d'utilisation.

Le but de notre étude est de répertorier les technologies et techniques d'interaction existantes pour saisir du texte. La présentation choisie, proposée en figure 1, regroupe les claviers en fonction de la technique d'interaction principale mise à disposition de l'utilisateur pour manipuler le clavier. Il est à noter que certains claviers combinent plusieurs de ces techniques. Ce classement nous permet de décrire une grande partie des claviers existants. Nous distinguons deux techniques d'interactions dominantes utilisées pour la saisie de texte : les interactions utilisant des touches et celles utilisant des gestes.

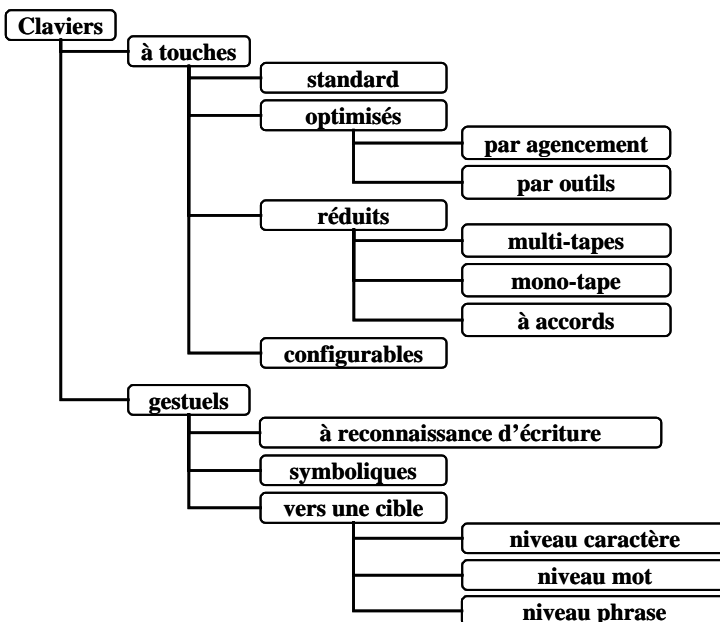


Figure 1. Classement des claviers en fonction de l'interaction dominante de saisie de texte

Dans la suite de ce document, chaque catégorie sera définie puis illustrée par différents claviers. Nous concentrerons essentiellement notre analyse sur les critères suivants : la présentation, le type de saisie, la prédiction, le contexte d'utilisation. Il convient avant tout de montrer par ces quelques critères non exhaustifs les avantages et les inconvénients des claviers présentés (par exemple, la simplicité d'apprentissage). Les performances des claviers seront quant à elles globalement présentées en synthèse des deux grandes catégories de claviers (les claviers à touches et les claviers gestuels).

La présentation représente l'aspect du clavier pour l'utilisateur. Il s'agit de décrire pour chaque type de clavier sa représentation physique et/ou virtuelle dans l'état initial et en cours de saisie. Ainsi pour chaque type de clavier, nous indiquons le type de retour pour l'utilisateur lors de sa saisie.

Le type de saisie permet de comprendre comment l'utilisateur parvient à saisir du texte. Il s'agit d'identifier les techniques d'interaction mises en œuvre. Même si c'est principalement ce critère qui définit notre classification, les techniques d'interaction non dominantes mais complémentaires seront également décrites. Ces dernières permettent souvent de particulariser certains claviers et améliorer leurs performances.

Le critère de prédiction n'est pas un critère de classement en soi car tous les claviers dont nous allons parler peuvent être associés à un système de prédiction de texte, dans le but d'accélérer la vitesse de saisie. Par contre, certains claviers intègrent systématiquement de tels systèmes, ce qui peut modifier sensiblement les performances lors de la saisie par rapport à d'autres claviers qui n'en utilisent pas automatiquement. (MacKenzie, 2002b) définit le rôle de ces systèmes. Ils apportent l'avantage d'accélérer la saisie grâce à un dictionnaire ; mais en contrepartie ce dictionnaire, qui sert de base à la prédiction, doit être adapté en fonction du langage, voire du domaine d'application. L'utilisateur a la charge supplémentaire de le compléter à chaque fois qu'un mot n'y est pas. Le retour visuel doit être de bonne qualité car le texte prédit doit être vérifié par l'utilisateur pour chaque lettre tapée d'où une saisie inconfortable (contrôle visuel fréquent) ; or cela ne s'avère pas toujours possible en fonction de l'écran. Globalement, deux techniques sont utilisées actuellement : la prédiction de mots et la prédiction de lettres. La première consiste à suggérer le mot ou la fin du mot en se basant par exemple sur le dictionnaire et la fréquence d'apparition. La seconde consiste soit à proposer la lettre suivante, soit à mettre en évidence les lettres probables, en utilisant par exemple un calcul sur les n-gram. Ces techniques peuvent être plus évoluées.

Le contexte d'utilisation est similaire à la notion de contexte d'interaction de (Poirier et Schadle, 2004) : bureau, véhicule, en mouvement, etc. Ce critère permet de révéler des spécificités importantes : les limites des claviers pour des utilisateurs à besoins spécifiques, les choix techniques et technologiques ou encore par exemple des informations sur le public d'utilisateurs visé (personne handicapée, personne en situation de mobilité, ...).

Les performances des différents claviers sont indiquées en mots par minute (ou wpm en anglais pour word per minute). Une telle unité de mesure pose une inconnue supplémentaire, qui est la taille du mot servant à la mesure. Ce problème est réglé par (MacKenzie, 2002a), qui exprime une convention communément admise pour les protocoles de tests. Celle-ci fixe la taille d'un mot à 5 caractères, incluant les espaces et autres tabulations, permettant ainsi une correspondance avec une mesure en caractères par seconde (CPS) : vitesse WPM = (60/5)*vitesse CPS. Les performances indiquées sont données à titre indicatif, mais sont difficilement comparables, car elles proviennent d'expérimentation qui ne se basent pas

forcément sur les mêmes protocoles (Poirier et Schadle, 2004). En particulier, des différences notables apparaissent dans les résultats des tests lorsque le protocole prend en compte ou non la correction des erreurs de saisie par l'utilisateur. Il est important de le souligner mais ce point n'est pas traité en priorité dans cet article.

De plus en plus d'appareils informatiques sont dépourvus de claviers physiques, pour des raisons de portabilité (assistants numériques) ou bien de sécurité et de maintenance (bornes automatiques SNCF). Les techniques de saisie utilisées donnent naissance à de nouveaux types de claviers réparties dans nos deux catégories principales de classement : les claviers logiciels. On désignera de manière générique par clavier logiciel tout système logiciel de saisie, quelle que soit sa forme, destiné à se substituer au clavier physique.

2 Claviers à touches

Les claviers à touches regroupent tous les claviers physiques ou logiciels qui utilisent le concept de touches pour saisir des lettres. De manière générale, quatre types de claviers se distinguent :

- Les claviers standard.
- Les claviers optimisés.
- Les claviers réduits.
- Les claviers configurables.

2.1 Claviers standard

De nos jours, les claviers de type Qwerty ou Azerty sont fortement majoritaires, et constituent donc un standard de fait. Ils peuvent être qualifiés de complets ou de non ambigus, dans le sens où chaque lettre est placée sur une touche distincte. Ils ne disposent pas par défaut de système de prédiction. Le contexte d'utilisation n'est pas spécifique : il reste pour un usage classique de bureau.

Les claviers Qwerty et Azerty sont apparus sur les machines à écrire, bien avant les premiers ordinateurs. La disposition des touches est due à un problème mécanique : les tiges portant les caractères se bloquaient lors d'un appui trop rapide sur deux touches proches. Sholes (Land, 1981) répartit les touches les plus utilisées de la langue anglaise de part et d'autre du clavier, concevant ainsi le premier clavier Qwerty en 1868 (figure 2). Le clavier Azerty est, quand à lui, une adaptation du clavier Qwerty à la langue française.

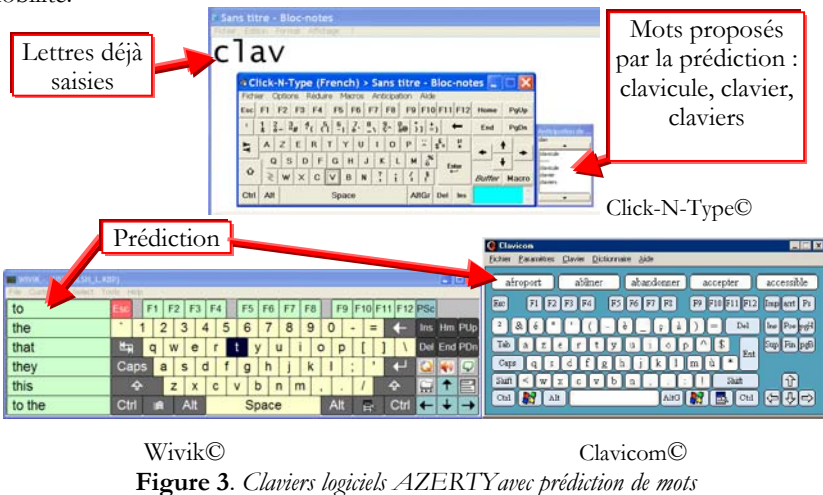
QWERTY KEYBOARD													AZERTY KEYBOARD														
~ 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	(9) 0	- +	=	Delete	1 &	2 é ~	3 #	4 { [(5 { [(6 - _	7 ` ~	8 \ /	9 ^ ª	0 ° +	=	←			
Tab	Q	W	E	R	T	Y	U	I	O	P	{	}	\	Tab	A	Z	E	R	T	Y	U	I	O	P	{	}	\
Caps	A	S	D	F	G	H	J	K	L	:	"	'	Enter	Caps	Q	S	D	F	G	H	J	K	L	M	;	'	Enter
Shift	Z	X	C	V	B	N	M	<	>	? /		Shift	Shift	>	W	X	C	V	B	N	?	/	;	'	Shift		
Ctrl		Alt									Alt		Ctrl	Ctrl		Alt							Alt		Ctrl		

Figure 2. Dispositions Qwerty et Azerty

Les claviers actuels d'ordinateur ont donc été conçus pour résoudre un problème mécanique qui n'a plus lieu d'être, aux dépens de l'ergonomie et de la vitesse de frappe. Depuis sa création, le clavier standard a évolué. Par exemple pour les ordinateur PC, le clavier est passé du format PC/XT au format PC/AT, puis étendu et enfin « compatible Windows », passant ainsi de 83 à 105 touches. Cependant, la disposition générale n'a pas évolué : ajout de touches d'accès à des

fonctions supplémentaires sans réelle réorganisation des lettres. Par exemple, le clavier de la société Belkin Corporation a été conçu pour améliorer le confort de frappe, et représente un modèle d'ergonomie pour un clavier commercial. Malgré cela, il reste au format Qwerty, ce qui est un handicap pour un clavier ergonomique car il occasionnera à long terme des douleurs, comme n'importe quel clavier standard (PCD Maltron LTD, 2004).

Les claviers standard existent sous une forme purement logicielle. Le clavier est affiché à l'écran et l'utilisateur doit sélectionner les touches virtuelles avec un pointeur. Copie logicielle conforme d'un clavier physique standard, ces claviers facilitent voire réduisent à zéro l'apprentissage nécessaire pour les habitués de l'ordinateur. Ils ont donc un avantage psychologique vis-à-vis de l'utilisateur, ce qui explique le nombre de claviers de cette sorte. Bien qu'ils respectent une disposition Qwerty/Azerty standard, beaucoup offrent des fonctionnalités supplémentaires. Ainsi les claviers ClaviCom©, WiVik© et Click-N-Type© (Ballayer, 2005) permettent de choisir le mode de sélection d'un caractère : par pointage et clic sur la touche, ou bien pointage et attente (paramétrable), ou encore par défilement. ClaviCom©, WiVik© et Click-N-Type© proposent une prédiction de mots, basée sur un dictionnaire que l'utilisateur peut compléter au besoin, et que Click-N-Type© peut compléter de manière automatique. Pour les trois claviers, les mots éligibles sont suggérés au cours de la frappe. En termes de performances, ce type de clavier est handicapé par la disposition Qwerty/Azerty adoptée (figure 3). Le contexte d'utilisation de ces claviers logiciels peut être étendu à des situations plus spécifiques que l'environnement bureau. Des utilisateurs à besoins spécifiques qui ne peuvent utiliser un clavier physique (notamment des personnes à motricité réduite) peuvent accéder à ces claviers. De plus, il peut être envisagé une utilisation en situation de mobilité.



Wivik© Clavicom©
Figure 3. Claviers logiciels AZERTY avec prédiction de mots

Une nouvelle génération de clavier tend à apparaître, les claviers projetés. L'utilisateur n'interagit ni sur des touches d'un clavier physique, ni sur des touches d'un clavier logiciel. Il interagit sur des touches projetées à l'aide d'un laser légèrement au-dessus d'une surface plane. Des capteurs détectent les intersections des doigts avec les rayons infrarouges. Un inconvénient majeur est la position des mains qui doivent rester au-dessus de la surface projetée. Une fatigue des doigts peut donc se ressentir. Ce type de clavier ne dispose pas de système de prédiction. (Roeber *et al.*, 2003) propose un clavier utilisant cette technologie (figure 4). Cela

permet de disposer d'un clavier de taille standard sur un périphérique de taille réduite, sous réserve de disposer d'une surface de projection suffisante. De plus, la matière de la surface de projection doit être bien choisie.



Figure 4. *Clavier à projection Canesta*

Tous ces claviers utilisent les dispositions classiques des caractères conçues de manière à éloigner les caractères les plus utilisés. D'où de nombreux et amples mouvements de curseur lors de la saisie, ce qui en ralentit la vitesse. D'après la loi de Fitts (Fitts, 1954), le temps mis pour atteindre une cible est proportionnel à l'index de difficulté qui dépend du \log_2 du ratio de la distance par la taille de la cible, ce qui engage à essayer certaines optimisations.

2.2 Claviers optimisés

Les claviers optimisés désignent des claviers à touches modifiées pour faciliter la saisie. On distingue deux types de modification : par agencement ou par outils. Notons que la prédiction de caractères ou de mots n'est pas un critère pour désigner un clavier optimisé dans cette partie. Afficher simplement les mots ou les caractères prédits n'optimise pas l'utilisation des touches du clavier, même si c'est une aide pour l'utilisateur. Par contre, l'interprétation éventuelle de ses résultats en termes de techniques d'interactions supplémentaires pour saisir les caractères permet de définir un clavier optimisé.

Claviers optimisés par agencement

Un clavier optimisé par agencement est un clavier physique ou logiciel qui utilise une répartition des caractères différente de celle des claviers standard afin d'augmenter la vitesse de saisie. Le contexte d'utilisation est *a priori* identique aux claviers standard.

Dès 1936, August Dvorak a mis au point un modèle de clavier physique qui porte son nom, pensé en termes de confort et d'efficacité. Le clavier Dvorak (figure 5) dispose, au contraire du clavier Qwerty/Azerty, les consonnes les plus utilisées et les voyelles sur la ligne médiane. À l'usage, il semble qu'il soit beaucoup plus confortable que le clavier Qwerty, car les mouvements des doigts seraient réduits et moins « acrobatiques » (Isokoski, 1999). L'histoire du clavier Dvorak (Brook, 2000) est contestée tout autant que ses performances par les « pro-Qwerty ». Bien que plus confortable, le faible écart de performances du clavier Dvorak par rapport au clavier Qwerty ne justifiait pas une nouvelle formation longue et coûteuse des typographes. De plus, le clavier Dvorak est restreint aux pays anglophones car il a été optimisé pour la langue anglaise.

DVORAK KEYBOARD

	@	#	\$	%	^	&	*	{	}	-	=	Delete
1	2	3	4	5	6	7	8	9	0			
Tab	?	<	>	P	Y	F	G	C	R	L	{	
	/	,	.								}	\
	A	O	E	U	I	D	H	T	N	S	"	Enter
Shift	:	Q	J	K	X	B	M	W	V	Z	'	Shift
	;										,	
Ctrl		Alt								Alt		Ctrl

<http://www.computerhope.com>

Figure 5. Disposition Dvorak

Un peu plus tard, une autre disposition fut mise au point dans le même but sous le nom de DIATHENSOR, correspondant aux dix lettres les plus utilisées de la langue anglaise (Pratt, 1939). Ces lettres sont placées dans cet ordre sur la ligne médiane du clavier. Cette disposition semble être plus ou moins abandonnée, au point même qu'on trouve très peu d'informations à propos de celle-ci.

En 2003, une nouvelle disposition de clavier est apparue : le clavier XPeRT (XPeRT, 2003). Contrairement au clavier Dvorak, il reste très proche d'un clavier Qwerty (figure 6) : il est donc plus facile à apprendre pour des utilisateurs de claviers standard. Le principe est simple : il s'agit de déplacer les couples de lettres les plus fréquentes (AS, ER, ON, TE) afin que chaque lettre d'un couple soit accessible par une main distincte pour accélérer la saisie. De plus, la lettre la plus fréquente (E) apparaît deux fois sur le clavier pour être accessible par les deux mains.



Figure 6. Disposition XPeRT

Partant du constat de l'inadéquation des claviers logiciels standard avec la loi de Fitts, des recherches sur l'agencement des claviers logiciels ont été également menées dans une optique unique de maximiser la vitesse de saisie en minimisant les déplacements du pointeur. La loi de Hick (Hick, 1952) et Hyman (Hyman, 1953) qui décrit le temps qu'il faut à un utilisateur pour prendre une décision en fonction du nombre de choix à sa disposition (la réaction en réponse à un message prend un temps proportionnel à l'information extraite du message qui elle-même augmente avec le nombre de messages possibles) a également servi de base pour améliorer certains claviers. Les solutions apportées sont plus ou moins fixes dans la mesure où elles utilisent un système de prédiction permettant un réagencement dynamique.

Parmi les nouvelles dispositions fixes étudiées dans ce but, le modèle OPTI (figure 7) publié par (MacKenzie et Zhang, 1999) en 1999, a été optimisé pour la langue anglaise et a pour particularité d'être muni de quatre barres d'espacement. Les dix lettres les plus utilisées sont placées au centre du clavier, et les autres plus ou moins au hasard (Zhai *et al.*, 2000). Les chercheurs ont retenu plusieurs dispositions qu'ils ont ensuite testées pour trouver la plus efficace. Un deuxième modèle, OPTI-II, comporte plus de touches au centre (figure 7). Une étude théorique réalisée par IBM *a posteriori* semble montrer que cette disposition est plus performante encore que la première.

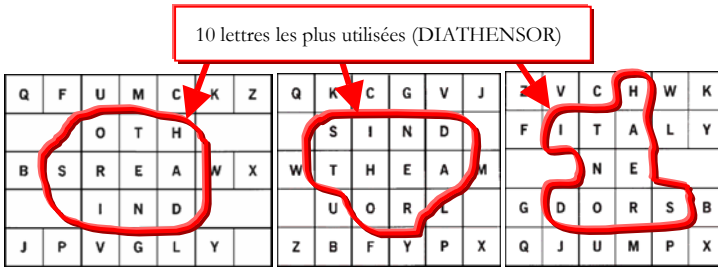


Figure 7. Dispositions OPTI (à gauche), OPTI-II (au milieu) et Fitaly (à droite)

Le modèle commercial Fitaly (figure 7) repose sur le même principe, mais il utilise deux touches d'espacement au lieu de quatre. Paru en 1998, donc antérieur au modèle OPTI, il est d'après l'étude (Zhai *et al.*, 2002) un peu moins rapide.

Les touches d'espacement sont multipliées afin de rapprocher l'espace des autres caractères, ce qui permet de minimiser les coefficients de la loi de Fitts dès la conception du clavier. Les trois dispositions précédentes tirent partie d'une spécificité de l'écrit anglais, où l'espace est un caractère fréquent, pour tenter de gagner en efficacité. Cela restreint d'autant leur domaine d'usage. Cependant, en plaçant les lettres les plus utilisées au centre du clavier, ils utilisent un principe direct de la loi de Fitts : plus les lettres que contient un mot sont proches sur le clavier logiciel, plus le mot en question sera saisi rapidement. IBM a repris ce principe pour créer une autre disposition dite Metropolis (Zhai *et al.*, 2000) (figure 8). Cette disposition porte le même nom que l'algorithme glouton qui a servi à la concevoir. L'initialisation de l'algorithme consiste à disposer les caractères aléatoirement dans un espace 2D (figure 9 à gauche), et les relier deux à deux par un lien d'attraction. Ce dernier est déterminé par les caractéristiques du langage. Il est d'autant plus fort que la probabilité que les caractères qu'il relie soient tapés consécutivement est élevée.

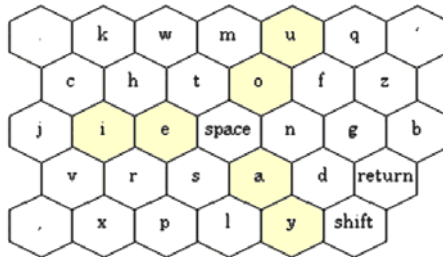


Figure 8. Un clavier Metropolis II

Ensuite, lors de la phase d'exécution de l'algorithme, les caractères vont progressivement se rapprocher de manière à minimiser la force d'attraction globale, résultant de la somme des attractions de tous les liens qu'ils portent (figure 9 à droite). On obtient ainsi une disposition très optimisée en termes de proximité des caractères fréquents. La figure 9 à droite présente la disposition des caractères pour une exécution de l'algorithme, au stade final. S'ils la jugent performante, les chercheurs peuvent ensuite décider de l'implémenter dans un clavier ; sinon ils peuvent aussi la rejeter. Les performances de chaque clavier ainsi conçu sont globalement identiques, mais toutes supérieures à celles de l'OPTI-II. L'avantage de cet algorithme est qu'il permet de créer en un temps réduit un clavier logiciel aux performances optimisées, quelle que soit la langue à laquelle ce dernier est destiné. Il

suffit en effet de préciser les caractéristiques du langage (principalement les fréquences des digrammes, apparition des différentes lettres après une lettre donnée) puis de lancer l'algorithme pour obtenir un clavier Metropolis. Cet avantage est aussi un inconvénient : pour concevoir un clavier optimisé pour un langage, il est nécessaire de connaître ou de déterminer un certain nombre de valeurs statistiques sur le langage cible.

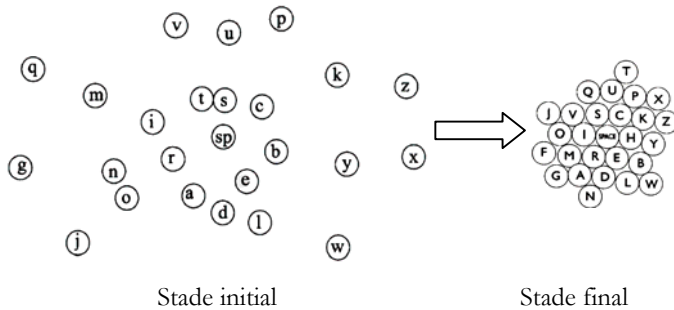


Figure 9. Algorithme Metropolis

Profitant de la possibilité de générer automatiquement des dispositions optimisées en fonction des contraintes, des caractéristiques supplémentaires sont ajoutées au modèle (Zhai *et al.*, 2002) afin de tenter d'améliorer l'apprentissage par l'utilisateur novice. Compte tenu que ce dernier passe plus de temps à chercher la touche qu'à effectuer le mouvement pour l'atteindre, ces contraintes concernent « l'optimisation alphabétique » (proximité relative de deux lettres consécutives de l'alphabet sur le clavier), la « connectivité des mots » (degré d'adjacence des lettres consécutives d'un mot sur le clavier) (Lee et Zhai, 2004) ainsi que d'autres améliorations ergonomiques. Les dispositions Metropolis obtenues satisfaisant l'ensemble des trois critères (optimisation de la disposition pour la performance, optimisation alphabétique et connectivité des mots) sont appelées dispositions ATOMIK (Alphabetically Tuned and Optimized Mobile Interface Keyboard). La figure 10 montre une implémentation adaptée au Palm.



Figure 10. Clavier Atomik sur Palm

Le clavier logiciel Atomik peut ainsi être utilisé en situation de mobilité. Il s'utilise avec un autocollant, idée initiée auparavant par (Softava, 2000) et (Krakow, 2000). Cet autocollant est fixé sur la zone du Palm réservée normalement au système de saisie Graffiti. Cette zone est entièrement calibrée par le logiciel pour se conformer à la position de l'autocollant : le clavier Atomik se substitue entièrement à celui de Graffiti. Il ne prend donc pas de place à l'écran par rapport à un affichage de clavier. Il est ainsi possible de profiter d'un système de saisie étudié pour la

performance (33% plus rapide que le Qwerty d'après (Zhai et al., 2002)) et pour le confort sur son Palm. De nombreux travaux continuent dans cette voie avec notamment l'utilisation d'algorithmes génétiques (Raynal et Vigouroux, 2005b). De manière théorique, les claviers GAG (Généré par Algorithme Génétique) ainsi obtenus permettent d'envisager des gains de performance de 50% pour une interaction au stylet (Raynal, 2006a).

Les dispositions nouvelles proposées peuvent aussi être dépendantes d'un contexte. C'est le cas des claviers logiciels utilisant un système prédictif qui réorganise les lettres au cours de la saisie. La méthode FOCL (Fluctuating Optimal Character Layout) (Bellman et MacKenzie, 1998) est utilisée sur des périphériques petits essentiellement en situation de mobilité. Elle consiste à réorganiser les lettres à chaque fois qu'une lettre est sélectionnée pour que les plus probables soient le plus près possible du pointeur. Le pointeur est déplacé via les flèches du clavier physique. La méthode FOCL réduit la moyenne de nombre de touches pour chaque lettre inscrite. Mais elle exige de l'utilisateur de trouver chaque lettre dans une disposition peu familière et qui varie d'où une perte de temps pour faire la recherche visuelle. Un test comparatif avec une disposition QWERTY fixe n'a pas montré de différences significatives entre les vitesses de saisie.

Le système Sybille de (Poirier et Schadle, 2005) utilise une approche similaire au système FOCL (Bellman et MacKenzie, 1998) pour les personnes handicapées d'origine cérébrale n'utilisant qu'un bouton poussoir comme interface d'accès (Boissière et Schadle, 2006). Il optimise son clavier en intégrant une prédiction de lettres et une prédiction de mots. À chaque lettre saisie, le clavier propose en premier les lettres les plus probables par rapport au contexte et dans une autre zone propose les mots les plus probables. Ainsi sur la figure 11, le clavier se décompose en plusieurs claviers : au milieu en haut le clavier propose les lettres les plus probables ('d', 'l', 'p', ...) et à gauche il liste les mots les plus probables ('le', 'la', 'et', 'de', 'un'). L'agencement des caractères est défini et mis à jour interactivement par le système de prédiction au cours de la saisie. Les touches du clavier permettent de saisir des lettres ou directement des mots.



Figure 11. Clavier Sybille

De manière générale, les claviers optimisés par agencement imposent un contrôle visuel lors de la saisie, car ils n'utilisent pas les configurations standard que les utilisateurs ont l'habitude d'utiliser. Même si certains apportent des améliorations notables aux modèles existants tant en termes de confort que d'efficacité, ils ne seront toujours pas adaptés à tous les utilisateurs en particulier pour ceux désirant prendre rapidement des notes sans contrôle visuel. Un apprentissage est nécessaire

pour acquérir des automatismes, mais les utilisateurs ne sont pas nécessairement prêts à le faire.

Claviers optimisés par outils

Un clavier est dit optimisé par outils si de nouvelles techniques d'interactions visuelles ou non sont implémentées pour améliorer la saisie sur les touches. Seuls les claviers logiciels sont concernés. La sélection des touches se fait à l'aide d'un pointeur contrôlable par la souris ou un stylet.

Le clavier Fisheye de (Raynal et Truillet, 2006) est affiché en miniature pour tenir sur une surface d'affichage réduite et utilise une technique de type « FishEye » (figure 12) pour avoir un agrandissement lors de l'interaction. Ce type de clavier optimisé est utilisé en situation de mobilité sur des périphériques réduits. Dès l'entrée en contact du stylet avec l'écran, les caractères autour du stylet sont agrandis. Le dernier caractère sur lequel se trouvait le stylet est validé lorsque le stylet est relevé.



Figure 12. Clavier passif et clavier actif Fisheye à proximité de la lettre f.

Afin de diminuer le chemin pour atteindre la touche désirée sur un clavier qwerty, (Jhaveri, 2003) propose d'accéder aux voyelles et à l'espace autour de chaque touche principale. L'idée a été approfondie par (Isokoski, 2004) en couplant un menu radial à un clavier QWERTY (figure 13). Ce menu permet de saisir une lettre supplémentaire seulement. Le principe est de pouvoir saisir deux caractères (digramme) en effectuant un seul geste sur une touche. L'auteur a choisi d'utiliser toujours le même menu avec un jeu de caractères fixe (les voyelles essentiellement) afin de faciliter l'apprentissage des caractères disponibles par menu. Lorsque l'utilisateur désigne une touche ('P' sur la figure 13), le caractère est validé. L'utilisateur peut à ce moment faire une sélection radiale pour entrer un nouveau caractère disponible dans un menu radial affiché sur le clavier logiciel.

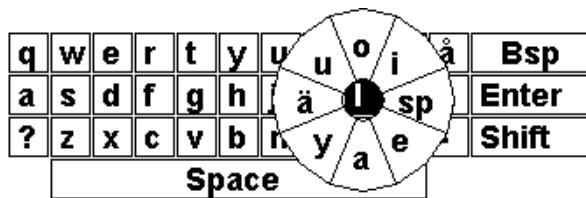


Figure 13. Le clavier QWERTY avec un menu radial

Un modèle théorique d'expérimentation indique des gains pouvant varier de 11 à 37%. Pourtant des tests ont montré que pour un utilisateur novice, la solution avec menus pénalisante au départ devient performante après 20 sessions de 15 minutes. L'auteur indique que la courbe de progression montre que la solution avec menus deviendrait plus rapide avec plus d'entraînement. Le contexte d'utilisation est *a priori* celui d'une utilisation classique (bureau). L'auteur précise qu'une utilisation sur un périphérique réduit (type assistant personnel) nécessiterait une étude supplémentaire quant à l'espace occupé par le menu.

Sur le même modèle, KeyGlass (Raynal et Vigouroux, 2005a) propose l'ajout de quatre touches supplémentaires semi-transparentes afin de ne pas trop

encombrer l'écran. Ces touches sont accessibles comme les autres touches par simple désignation. Par contre, le choix des caractères disponibles se fait à l'aide d'un système de prédiction au contraire de (Isokoski, 2004). Une première expérimentation montre que l'ajout du système KeyGlass (figure 14) diminue bien les distances de sélection mais cette même expérimentation montre que la vitesse d'écriture diminue également de façon significative (Raynal, 2006b). Le fait que les caractères changent en fonction de la prédiction ressort comme le principal responsable de cette chute de performance.

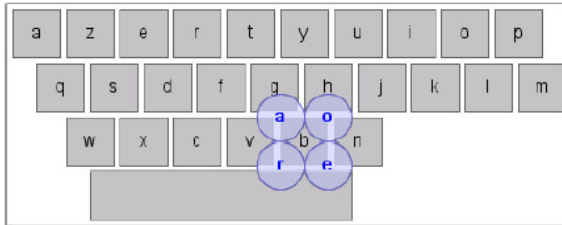


Figure 14. KeyGlass : disposition des touches après avoir saisi la lettre 'b'.

Parmi les claviers utilisant la prédiction pour introduire une technique d'interaction, la proposition basée sur un clavier AZERTY de (Magnien *et al.*, 2004) permet d'aider la saisie pour les utilisateurs novices ou dans des conditions où l'interaction est dégradée en particulier en situation de mobilité. L'idée consiste à introduire des indices visuels en mettant en contraste les touches susceptibles d'être sélectionnées (figure 15). (Magnien *et al.*, 2004) montre que cette technique a l'avantage de s'adapter sur de nombreux claviers (Azerty, Metropolis). Une étude empirique de (Magnien *et al.*, 2003), basée sur la loi de Hick-Hyman montre un gain théorique de 50% pour un usager novice. (Magnien *et al.*, 2004) montre une utilisation sur un assistant personnel où les caractères sont répartis aléatoirement pour réaliser des tests avec des novices. Les résultats donnent un gain d'environ 60% dans les mêmes conditions de test sans indice visuel.



Figure 15. Indices visuels pendant la saisie du mot « interaction »

La prédiction de mots utilisée par le système POBox de (Masui, 1999) ajoute dynamiquement des touches représentant les mots probables par rapport aux lettres saisies (figure 16). Une version précédente affichait l'ensemble des mots dans un menu contextuel de boutons affiché à côté de la lettre saisie. Ce clavier est destiné à un usage en situation de mobilité, partout où un clavier classique ne peut être utilisé simplement. Les mots probables affichés permettent une saisie plus facile. Il est toutefois difficile d'envisager ce système sur des périphériques réduits en situation de mobilité tant les mots prennent de la place. Aucun test utilisateur n'a été effectué à ce jour.

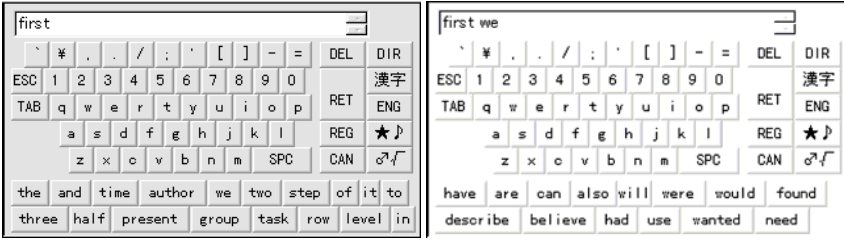


Figure 16. Système POBox

Le contexte d'utilisation des jeux vidéo et en particulier des consoles de jeux requièrent une toute autre approche pour la saisie de texte. L'utilisateur doit utiliser un contrôleur de jeu pour saisir du texte : classiquement un joystick du contrôleur est utilisé, ce qui n'est pas simple lorsque le texte est long. Une approche consiste à conserver un clavier standard QWERTY et à utiliser une technique d'interaction bi-manuelle avec un contrôleur de jeu à deux joysticks (figure 17). L'intérêt est de conserver l'expérience des utilisateurs pour les claviers standard. Chaque joystick permet de sélectionner une moitié du clavier. Deux études similaires ont été réalisées par (Wilson et Agrawala, 2006) et (Sandnes et Aubert, 2007).

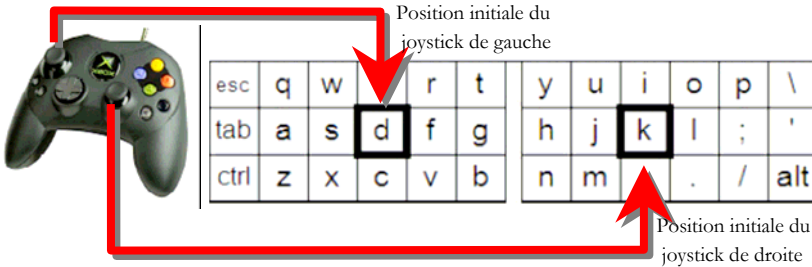


Figure 17. Clavier bi-manuel à deux joysticks

2.3 Claviers réduits

Le contexte d'utilisation ne permet pas toujours d'envisager un clavier standard ou optimisé. La place requise pour ces claviers n'est pas toujours disponible ou bien ils ne sont pas toujours adaptés aux utilisateurs. Par exemple, de nombreux problèmes se posent lors de l'utilisation des claviers avec des périphériques mobiles, car ils disposent d'une surface réduite rendant impossible la conception de claviers de près de cent touches ainsi que l'utilisation de plusieurs doigts pour la frappe. Une personne handicapée peut également éprouver des difficultés à accéder à toutes les touches d'un clavier. Ainsi, les claviers réduits ont été pensés. Ils désignent tous les claviers à touches qui occupent un minimum de surface par rapport aux claviers à touches standard. On distingue trois types de claviers réduits en prenant comme critère l'interaction nécessaire sur les touches du clavier pour accéder à une lettre :

- Les claviers multi-tapes.
- Les claviers mono-tapes.
- Les claviers à accords.

De par leur taille réduite, certains claviers auront des touches qui donneront accès à plusieurs caractères : ces claviers seront dits « ambigus » (Poirier et Schadle, 2004). Par opposition un clavier réduit non ambigu permet de déterminer un unique caractère par l'appui d'une ou plusieurs touches sans nécessité de mécanisme de désambiguïsation supplémentaire.

Claviers multi-tapes

(Pavlovych et Stuerzlinger, 2004) présente les claviers multi-tapes comme des claviers où on accède à un caractère par plusieurs appuis successifs sur une même touche. Cette définition, jugée trop réductrice, est étendue à plusieurs touches. Ainsi, dans cet article, un clavier multi-tapes est un clavier où on accède à un caractère par plusieurs appuis successifs sur au moins une touche. Un caractère est défini par une séquence ordonnée de touches. Cette technique permet de réduire le nombre de touches. Par contre, ce type de clavier est ambigu car une touche donnée permet d'accéder à plusieurs caractères : c'est une séquence ordonnée de touches qui permet de lever l'ambiguïté.

Certainement le plus connu des claviers réduits multi-tapes, le clavier de téléphone mobile standard à douze touches est présent sur la plupart des téléphones (figure 18). Les caractères accessibles par une touche sont affichés sur la touche. Par nature, la vitesse de saisie sur un clavier à douze touches est lente. Malgré cela, il reste le plus utilisé, et il est généralement associé à un algorithme de prédiction afin d'accélérer la saisie.



Clavier standard à douze touches



Clavier MultiPlay

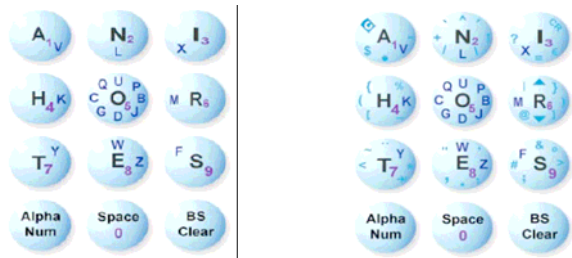
Figure 18. Claviers de téléphones mobiles

Dans le but toujours d'accélérer la vitesse de saisie, des évolutions du clavier de téléphone mobile à douze touches voient le jour. Le système RollPad (Oniszczak et MacKenzie, 2004) propose de conserver l'agencement initial mais remplace les touches par une surface tactile. L'interaction est basée sur la sélection d'une touche puis d'un léger roulement ou glissement du doigt pour sélectionner le bon caractère. Par exemple un appui sur la touche 3 génèrera le caractère E mais un appui sur la touche 3 suivi d'un roulement du doigt vers la gauche génèrera le caractère D. Cet appui suivi d'un roulement s'apparente à un clavier multi-tapes. Les tests menés ne montrent pas de différences significatives avec le clavier à douze touches standard hormis le nombre d'appuis par caractère.

Le système MultiPlay® (KeyTouch, 2003) propose des modifications plus profondes : les lettres sont disposées aux quatre coins cardinaux des touches. Une croix directionnelle de navigation est également ajoutée, c'est-à-dire un bouton à quatre directions permettant de choisir et de valider le caractère (figure 18). Le système porte le nombre minimum de gestes par caractère à 2, car il faut au moins choisir et valider la commande. En contrepartie, il offre l'avantage d'étendre le clavier à douze touches avec des fonctions additionnelles.

Le clavier MessageEase introduit par (Nesbat, 2003) est basé sur un clavier à douze touches pour être utilisable sur de petites périphériques (téléphones mobiles,

montres, tablet PC,...). La répartition des lettres est différente d'un clavier multi-tapes. Les lettres sont réparties par rapport à leur fréquence d'utilisation. Le concept est simple : une lettre très fréquente doit être accessible rapidement. La figure 19 montre la répartition adoptée : les neuf lettres les plus fréquentes sont réparties au centre des touches (de n°1 à n°9) et sont accessibles par double clic. Huit lettres moins fréquentes sont réparties sur les mêmes touches sauf la touche centrale (n°5), mais disposées en périphérie et orientée vers la touche centrale. Elles sont accessibles par deux sélections de touches : la touche qui la contient puis la touche centrale. Par exemple, 'M' est obtenu en choisissant les touches 6 puis 5. Les huit lettres restantes sont réparties sur la touche centrale (n°5) et sont disposées de sorte que chaque lettre puisse désigner une touche autour de la touche centrale. Ces touches sont aussi accessibles par deux sélections de touches : la touche centrale, puis la touche désignée par la lettre. Par exemple, 'G' est obtenu en choisissant les touches 5 puis 7. En appliquant cette répartition sur les autres touches et en considérant le clavier circulaire, on peut étendre le clavier à d'autres caractères (figure 19 à droite en haut). Un calcul théorique montre que ce clavier atteint des performances légèrement supérieures au clavier multi-tapes de type téléphone à douze touches.



Répartition des lettres

Répartition avec d'autres caractères

Figure 19. Clavier MessageEase

Concernant les claviers logiciels, on retrouve les versions logicielles des claviers physiques réduits multi-tapes (MacKenzie et Zhang, 1999). Par exemple, (Vella et Vigouroux., 2004) propose une version logicielle du clavier physique réduit type téléphone mobile avec et sans prédiction. L'objectif était de le comparer en termes de performances avec un clavier logiciel standard et ce, sur une population valide et une population handicapée. Le fait marquant de cette étude est certainement que, sur la population testée, les clics successifs sur le clavier ambigu semblaient plus difficiles à réaliser que les mouvements du pointeur sur le clavier standard.

(SpeedScript, 1999) propose une réduction de taille moins drastique : les voyelles et l'espace sont associés à la même touche, de même que les caractères x, y, z peu fréquents lors de la frappe. Pour entrer une consonne, il suffit d'appuyer sur la touche correspondante. Pour entrer une voyelle, un appui sur la touche des voyelles fait apparaître une fenêtre bleue et l'utilisateur doit alors faire un geste vers la voyelle désirée (figure 20). Cette technique est étendue aux consonnes, en maintenant le pointeur sur la consonne, la même fenêtre bleue apparaît permettant de saisir des digrammes à la manière de (Isokoski, 2004) sur un clavier standard. Il est même possible de continuer d'autres voyelles directement en continuant à dessiner des gestes dans la direction voulue. Contrairement au clavier de (Isokoski, 2004), il n'est pas possible d'accéder à toutes les lettres par un seul appui sur une touche : les voyelles par exemple ne peuvent être choisies que par deux appuis. Ce

clavier est bien dans la catégorie des claviers multi-tapes. À ce jour, aucune étude n'a été publiée sur ce clavier.

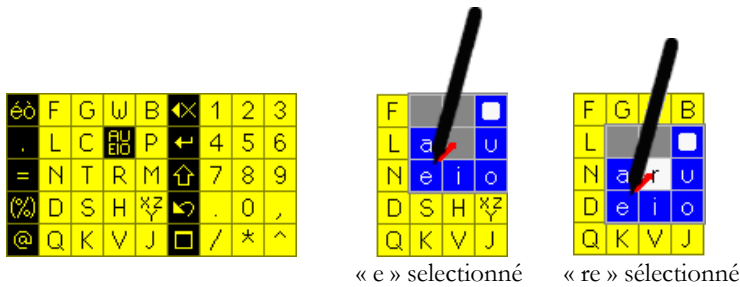


Figure 20. Clavier SpeedScript smart

Le clavier Half-QWERTY (Matias *et al.*, 1996) réduit un clavier QWERTY en le décomposant en deux parties. L'idée consiste à offrir un clavier standard QWERTY sur une seule main, gauche ou droite, selon la préférence de l'utilisateur. Les auteurs envisagent d'utiliser ce type de clavier en situation de mobilité sur périphérique réduit ou pour des personnes handicapées. L'architecture est choisie afin de faciliter le transfert de compétence de la frappe à deux mains vers la frappe à une main. L'agencement se base sur un demi-clavier QWERTY sur lequel est « repliée » l'autre moitié du clavier (figure 21). Lorsqu'une touche est relâchée, le caractère en haut à gauche de la touche est validé. Par contre, si la barre d'espace est appuyée au préalable, c'est le caractère en bas à droite de la touche qui est validé. Ce clavier correspond bien à un mode multi-tapes. Cette technique peut-être étendue avec des modificateurs, des touches particulières afin d'étendre le jeu de caractères (Shift, ...). Ce clavier présente l'avantage de la similitude avec le clavier QWERTY et semble offrir une bonne vitesse d'apprentissage.

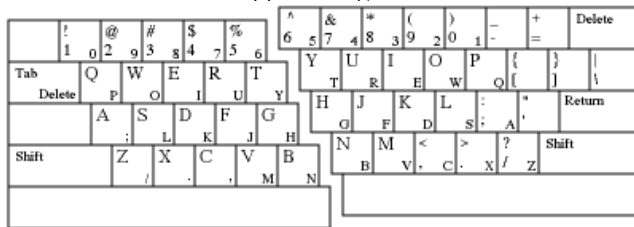


Figure 21. Agencements du clavier Half-QWERTY pour la main gauche et la main droite.

Le clavier DotNote (MacKenzie et Soukoreff, 2002) est un clavier logiciel basé sur de larges touches afin d'offrir une frappe à une main sur un assistant personnel : le pouce sélectionne les touches alors que l'assistant est tenu avec la même main. L'interaction peut faire appel à d'autres périphériques. Les touches étant très larges, il n'est pas possible d'afficher tous les caractères. Contrairement au clavier Half-Qwerty, l'idée est de répartir les lettres sur deux pages séparées, l'une seulement étant accessible à un instant donné (figure 22). DotNote est optimisé dans la mesure où les lettres les plus fréquentes apparaissent sur le clavier standard et une touche permet le basculement vers le clavier secondaire. Sur les deux demi-claviers, les lettres sont arrangées de manière alphabétique. Ce clavier est multi-tapes car pour accéder à certaines lettres, il faudra appuyer sur deux touches (celle pour changer de page, puis celle contenant la lettre désirée). Malheureusement aucune étude n'a été publiée sur ce clavier.

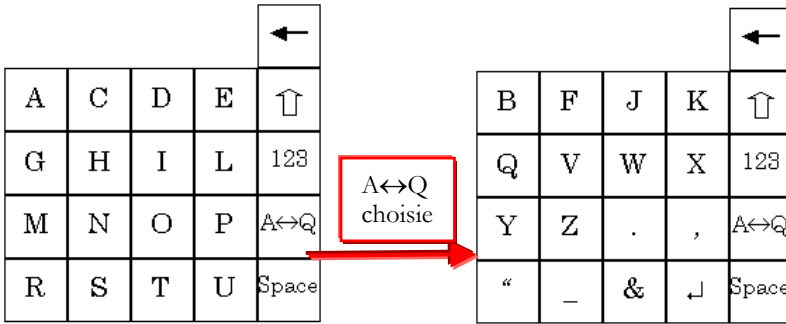


Figure 22. Clavier DotNote

Le clavier FrogPad (Carey, 2004) (figure 23) est un clavier multi-tapes où la séquence de touches définissant un caractère peut comprendre une touche de fonctions qui doit rester maintenue au moins pendant la séquence. Elle peut le rester au delà, ce qui permet d'accéder à d'autres caractères nécessitant également cette touche de fonction. Pour réduire le temps d'apprentissage des combinaisons de touches à effectuer pour atteindre un caractère, une méthode de désambiguïsation simple est mise en place : un retour visuel directement sur la touche des lettres sélectionnables, la position de la lettre sur la touche et sa couleur. De par la position et la couleur du caractère sur la touche, on sait automatiquement quelle combinaison on doit utiliser pour obtenir le caractère voulu. Ce mode de fonctionnement rassure : on utilise des touches de fonctions connues par l'utilisateur de clavier (CTRL, Shift, ALT, etc.) plutôt que des accords à mémoriser.



Figure 23. Claviers FrogPad

Le clavier In10Did (In10did, 2005) est basé sur 10 touches : une pour chaque doigt de la main. La touche n'est pas physiquement nécessaire : un gant de donnée peut être utilisé pour capturer la pression des doigts. L'idée de ce clavier est de taper sur des touches virtuelles ou non sans déplacer les doigts. La saisie est multi-tapes car une lettre est saisie par pression d'un ou deux doigts au maximum. L'ordre des doigts a de l'importance ainsi que le doigt lui-même. La figure 24 montre comment saisir les lettres A, P et Y. Un apprentissage est nécessaire même si la répartition des lettres sur les doigts tend à suivre l'ordre alphabétique. Le contexte d'utilisation est varié : situation de mobilité, assistant personnel, montre, téléphone, voiture, ... Ce clavier a été conçu pour limiter l'espace pris par la saisie de texte afin de pouvoir être utilisé dans un espace réduit.

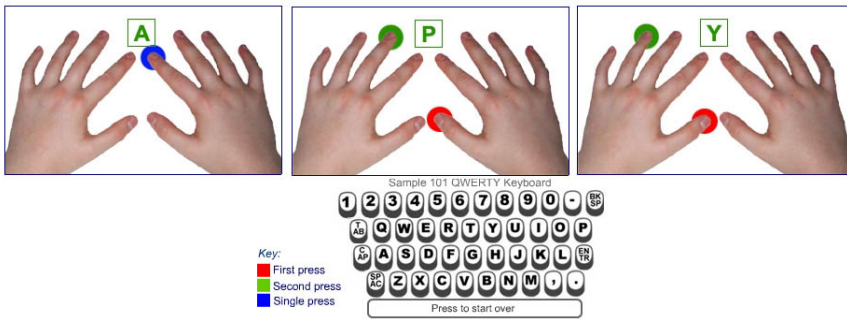


Figure 24. Clavier In10Did.

Claviers mono-tape

Un clavier mono-tape est un clavier réduit qui permet d'accéder à un caractère en sélectionnant une seule touche et en utilisant un système de désambiguïsation physique ou logiciel implicite ou explicite. Le type mono-tape ne signifie donc pas que l'utilisateur accèdera à un caractère par l'appui sur une touche, le système de désambiguïsation peut amener des interactions sur des touches spécifiques. Ainsi, le nombre moyen de touches par caractères, KSPC (keystrokes per character) (MacKenzie, 2002b), ne vaut pas forcément 1. On peut noter globalement des inconvénients communs aux claviers mono-tape utilisant un système de prédiction. Le système doit être adapté aux utilisateurs et au contexte d'utilisation (par exemple, SMS) pour obtenir de bons résultats. De plus, en cas d'erreur d'interprétation, la technique d'interaction pour corriger n'est pas toujours intuitive.

Sur la majorité des téléphones mobiles à douze touches, on trouvera le système T9® de Tegic Communications, qui réalise une prédiction de mots, ce qui réduit le nombre d'appuis successifs nécessaires pour appuyer sur une touche. La figure 25 donne un exemple : le mot « how » ne nécessite que l'appui unique sur chaque touche avec le système T9®.

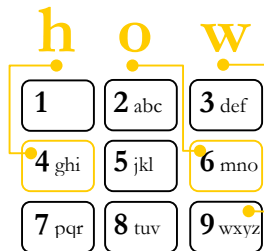


Figure 25. Utilisation du système T9® sur un clavier multi-tapes

D'autres algorithmes se développent, notamment chez Eaton Ergonomics avec WordWise® (Kober et al., 2001) pour la prédiction de mots, et LetterWise® (MacKenzie et al., 2001) ou PNLH (Gong et al., 2005) pour la prédiction de lettres. Ces types de claviers avec prédictions restent ambigus : une touche donne toujours accès à plusieurs caractères et la prédiction peut lever l'ambiguïté.

Afin d'améliorer encore les performances des systèmes de prédiction ou de faciliter la frappe, des travaux ont porté sur des agencements différents des lettres sur les touches. Basés sur les systèmes de prédiction LetterWise et WordWise, les claviers réduits EQx (Clavier QWERTY Eaton à x colonnes) proposent par exemple un agencement inspiré des claviers QWERTY. Cela permet de transposer

la connaissance des claviers standard à ces claviers réduits mais également de réduire les problèmes de prédictions (Eatoni, 2007). Ces claviers deviennent réduits et optimisés par agencement. Aucun test des claviers EQx n'a été publié à ce jour. (Gong et Tarasewich, 2005) présente également une expérimentation sur deux agencements : le premier n'impose aucune contrainte sur les lettres associées à une touche alors que le second impose que les lettres associées à une même touche soient consécutives dans l'alphabet (figure 26).

ABC	DEF
GHI	JKL
MNO	PQRS
TUV	WXYZ

AFLX	GPUY
MOQ	DEV
BIT	NWZ
HS	CJKR

ABCD	EFG
HIJKL	MN
O	PQRS
TUV	WXYZ

Figure 26. Agencements standard, optimisé non contraint et optimisé contraint pour clavier réduit optimisé par agencement

Les performances obtenues par des débutants avec ces deux claviers réduits optimisés (figure 26 au milieu et à droite) confirment qu'un agencement non contraint est pénalisé en performance. L'agencement contraint facilite la recherche de la lettre lors des premiers usages. Dès la seconde session (14 phrases par session), l'agencement contraint donne des performances comparables laissant envisager un gain après apprentissage. Les tests ne sont cependant pas allés au-delà des ces deux premières sessions. Des résultats similaires ont été obtenus pour LetterEase (Ryu et Cruz, 2005) même si dans ce cas, l'agencement non contraint proposé, semble obtenir de meilleurs résultats en nombre d'erreurs. Cependant le coût d'apprentissage des ces différents agencements ne semble pas être compensé par des gains substantiels en performance.

Pour les périphériques munis d'un écran tactile, le nombre de touches utilisables peut être plus important et les interactions plus évoluées. Le clavier TouchPal (CooTek, 2008) se base sur la technologie T+ qui combine un système de prédiction et la détection des mouvements du doigt. Le clavier reprend l'agencement d'un clavier classique ce qui devrait permet un apprentissage plus rapide (figure 27).



Figure 27. Utilisation du système TouchPal

L'appui sur une touche est ambigu dans la mesure où deux caractères sont candidats (annotation 1 de la figure 27). L'utilisateur peut toutefois lever l'ambiguïté immédiatement lors de la frappe par un simple glissé de son doigt, ce clavier devient alors mono-tape. Par exemple, un appui sur la touche 'gh' puis un glissé sur la gauche revient à sélectionner le caractère 'g' (annotation 2 de la figure 27). À ce jour, aucune étude de ce clavier n'est encore disponible.

Le clavier UKO-II (Unbekanntes Kommunikationsobjekt - objet de communication non identifié) (Harbusch *et al.*, 2003) utilise le même type d'interaction que les téléphones mobiles avec prédiction ; il est ambigu pour les mêmes raisons. La présentation et le contexte d'utilisation est différent. En se basant essentiellement sur une sélection par défilement, ce clavier tente de réduire au maximum le nombre de touches, nombre qui est modifiable.

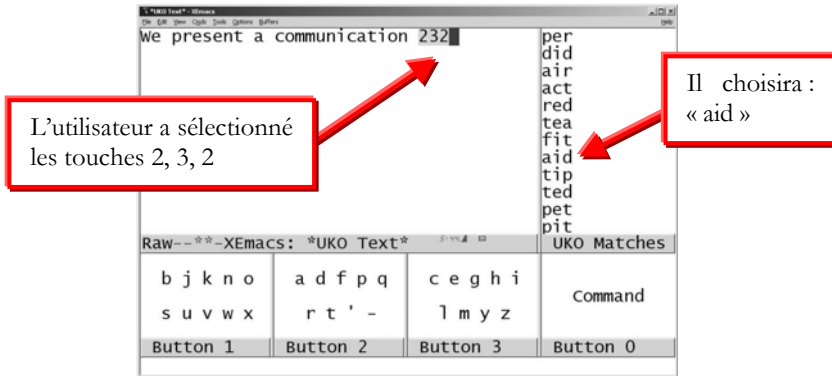


Figure 28. Le clavier UKO-II

La figure 28 montre une version UKO-II avec 4 boutons. Les agencements de lettres, plusieurs par touche, sont fixes et sont optimisés selon la langue de l'utilisateur. Pour choisir une lettre, l'utilisateur doit sélectionner le bouton qui la désigne. Pour la désambiguïsation, ce clavier s'appuie sur une prédiction de mots activable par l'utilisateur. Celui-ci peut choisir son mot dans une liste. Si le mot à saisir n'est pas dans la liste, il est possible de l'ajouter au lexique. Ce type de clavier est initialement conçu pour des utilisateurs atteints d'infirmité motrice cérébrale. La sélection des touches est d'ailleurs prévue par une technique de suivi du regard. La charge cognitive de l'utilisateur est importante quand il saisit un mot car il ne peut pas voir les lettres déjà saisies avant de sélectionner son mot dans la liste de choix (Evreinova *et al.*, 2004). Un mot est obtenu par une séquence de touches sélectionnées, complétée par un système de prédiction. En fonction de l'ordre des touches choisies et des lettres présentes sur les touches, le système de prédiction propose à l'utilisateur de choisir son mot dans une liste de mots probables.

Le système TiltText de (Wigdor et Balakrishnan, 2003) permet d'utiliser un clavier de téléphone mobile à douze touches en mode mono-tape. Le système de désambiguïsation utilisé est basé sur l'inclinaison du téléphone (figure 29). Lorsqu'une touche est sélectionnée, l'utilisateur doit ensuite incliner le téléphone dans l'une des directions (gauche, droit, avant, arrière) associée à la lettre désirée. Les performances sont légèrement supérieures à celle d'un clavier multi-tapes des téléphones à 12 touches. Les inclinaisons en avant et en arrière procurent le plus d'erreurs : la prise en main des téléphones souvent en avant peut expliquer les difficultés pour saisir les lettres nécessitant une inclinaison en avant ou en arrière.

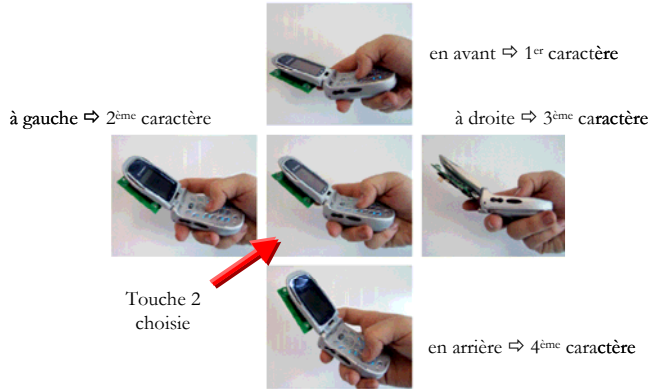


Figure 29. Le système TiltText

Le clavier MessagEase présenté en version physique multi-tapes (cf. figure 19) existe aussi en version logiciel mono-tape (Nesbat, 2003). La conception du clavier est identique ; l'interaction pour choisir les lettres est différente et simplifiée. Pour accéder aux lettres fréquentes, une seule sélection suffit. Pour accéder aux lettres nécessitant deux touches à sélectionner, il suffit de sélectionner la touche comportant la lettre, puis d'effectuer un court glissement vers la seconde touche. Cette version logicielle est disponible sur de nombreux supports (montre, téléphone, assistant personnel, tablet PC, ...).

Le clavier Claviature présenté par (Claviature, 2006) peut être comparé au clavier MessagEase dans sa version logicielle. Il s'agit d'un clavier logiciel où chaque touche contient quatre caractères aux quatre coins. La répartition des caractères dépend de la langue ainsi que de la fréquences des lettres ou couple de lettres dans la langue utilisée. La technique d'interaction est comparable dans la mesure où il faut sélectionner la touche qui contient le caractère désiré, puis la désambiguïser s'effectue par un glissement montrant dans quelle direction se trouve la lettre sur la touche. La figure 30 montre comment saisir la lettre 'q'. Ce type de clavier a l'avantage de pouvoir proposer de larges touches permettant une saisie au doigt. En situation de mobilité cette technique pourrait faciliter les interactions de l'utilisateur.



Figure 30. Le clavier Claviature : sélection de la lettre 'q'.

Le clavier Fastap® (Cockburn et Siresena, 2003) est un clavier mono-tape destiné aux téléphones portables. Il intercale les lettres sur des touches à part, portant le nombre de touches à près de 45. Le système de désambiguïser est simplement physique ici rendant le clavier non ambigu. En voyant la figure 31, on pourrait dire qu'il faut avoir les doigts fins pour se servir d'un tel clavier ; mais on pourrait de plus s'interroger sur la fiabilité, car multiplier les touches revient à multiplier le risque de défaillance de l'une d'entre elles. De plus la disposition des touches n'utilise ni les agencements standard ni les agencements optimisés, mais reprend simplement l'ordre alphabétique. Le but est d'offrir un accès rapide aux lettres sans apprentissage préalable. Le contexte d'utilisation est celui des téléphones

de dernière génération, qui permettent de lire les courriers électroniques et surtout d'y répondre de manière efficace. Les tests effectués ont montré que les utilisateurs étaient rapidement performants par rapport au clavier classique de téléphone utilisant le système T9 de prédiction de mots. Bien que le système T9 permette une saisie rapide de mots du dictionnaire après apprentissage, le clavier Fastap® est globalement plus rapide pour saisir tout type de caractères (mots du dictionnaire ou non, abréviations ou nombre). L'auteur précise qu'un système de prédiction pourrait améliorer les performances du clavier Fastap®.



Figure 31. Clavier Fastap

Plus simplement, des claviers standard ont été physiquement réduits pour tenir sur un téléphone mobile ou un assistant personnel. Ces claviers sont appelés des claviers mini-Qwerty. La figure 32 montre quelques modèles. Ils sont conçus pour une saisie du texte avec les deux pouces. (Clawson *et al.*, 2005) a montré qu'en situation d'interaction en aveugle (clavier non accessible visuellement ou non accès à l'écran de saisie) n'est pas adapté. Un apprentissage montre que les utilisateurs améliorent la précision en atteignant les chiffres en mode normal (retour visuel accessible), mais la vitesse de saisie n'augmente que très peu. L'atout des mini-Qwerty est l'accès rapide à la technique d'interaction pour la saisie.



RIM BlackBerry



Nokia 6820



Danger Hiptop/T-Mobile Sidekick



Targus



Dell

Figure 32. Claviers mini-Qwerty (Clarkson *et al.*, 2005)

Claviers à accords

Les claviers à accords associent un caractère à une combinaison de touches, pressées simultanément (Soukoreff, 2002). La simultanéité ne pouvant être physiquement garantie, l'ordre d'appui des touches n'a pas d'importance dans ce type de clavier. Contrairement aux claviers multi-tapes, les touches sélectionnées ne peuvent être maintenues d'un accord à l'autre. Ces claviers sont ambigus dès lors qu'une touche peut être utilisée par plusieurs accords. Le nombre de touches varie selon le clavier, et, pour un clavier à n touches, il est possible de coder $2^n - 1$ caractères.

Proposés en tant qu'extension à un système déjà existant (par exemple un assistant numérique ou un ordinateur ultra-portable) ou bien comme remplacement du clavier pour les personnes déficientes, les claviers à accords sont conçus pour autoriser la mobilité tout en conservant une vitesse de frappe soutenue. Malheureusement, ils nécessitent un apprentissage des combinaisons de touches, ou accords, avant de pouvoir taper rapidement ; ce qui peut paraître un obstacle (comme pour le clavier Dvorak). Mais les résultats donnent des performances proches des claviers standard si l'apprentissage est suffisant (Isokoski, 1999) et ils présentent l'avantage de pouvoir être utilisés à une main (Soukoreff, 2002) (Lyons *et al.*, 2004b). Plus problématique, les claviers à accords n'offrent pas de retour visuel, car ils sont destinés à une saisie en aveugle. On ne peut donc pas voir quel caractère on compose.

Celui que l'on considère comme le premier système proposant ce type de clavier est le système NLS (oN Line System) (Engelbart, 1968). Il utilisait en plus du clavier standard et d'une souris un ensemble de cinq touches, « chord key set » (figure 33). Le but était de permettre à l'utilisateur de réaliser des tâches ou saisir du texte avec une seule main en utilisant au maximum les cinq doigts pendant que l'autre main utilisait la souris. Ce clavier à accords est fortement inspiré du clavier à cinq touches du télégraphe Baudot où un caractère était codé sur 5 bits soit 32 possibilités auxquelles s'ajoutent 2 caractères spéciaux éventuellement, ce qui offre 64 possibilités (26 caractères pour l'alphabet, 10 pour les chiffres, plus quelques caractères pour la ponctuation) (Thomasset, 1999).

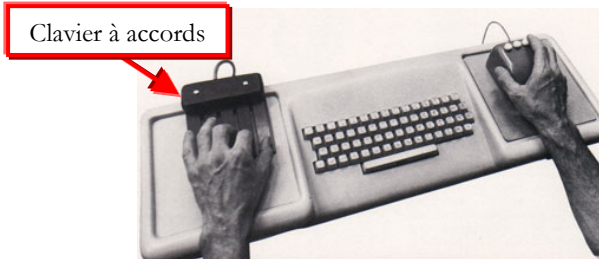


Figure 33. Clavier à accords du système NLS

Le clavier CyKey (figure 34) (Bellaire, 2006), dernière génération du Microwriter est un clavier à accords qui utilise le système de touches à accords Microwriting de (Endfield, 1978). Le codage des symboles se fait sur cinq touches, soit trente symboles possibles. Le clavier CyKey dispose de quatre touches de fonction, portant le nombre de possibilités à 150, ce qui permet de coder l'ensemble des touches d'un clavier standard. Mais si le système Microwriting a eu le temps de faire ses preuves en vingt ans d'utilisation, le modèle de clavier CyKey n'est pas vraiment fait pour être associé à un assistant numérique ou bien un téléphone portable. Les concepts même de la méthode font que le périphérique doit pouvoir

se placer à plat, sous la main, d'où un encombrement conséquent, aussi réduit le clavier soit-il.

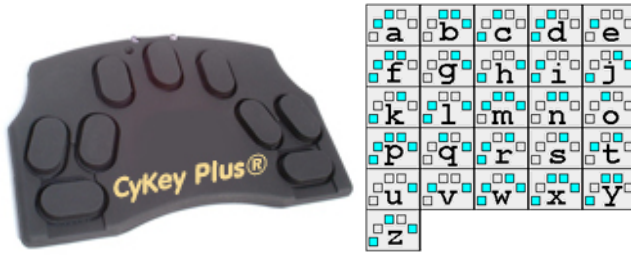


Figure 34. Clavier CyKey et sa table de caractères

Il existe d'autres claviers destinés spécifiquement aux appareils ultra-portables. Ainsi, le GKOS (Global Keyboard Optimized for Small wireless devices) (Tiainen, 2000) est un clavier à six touches conçu pour se placer directement dans le dos d'un périphérique portatif (figure 35). Pour l'instant, le clavier GKOS reste à l'état de standard ouvert, même si ses concepteurs continuent de proposer différents modèles et usages du concept. Le GKOS permet de coder 62 caractères (sur six touches), soit un alphabet complet avec les caractères accentués et la ponctuation. Contrairement à l'approche classique des claviers à accords, il a été conçu pour être utilisé avec les deux mains, même si une utilisation avec une main reste possible quoique moins performante.

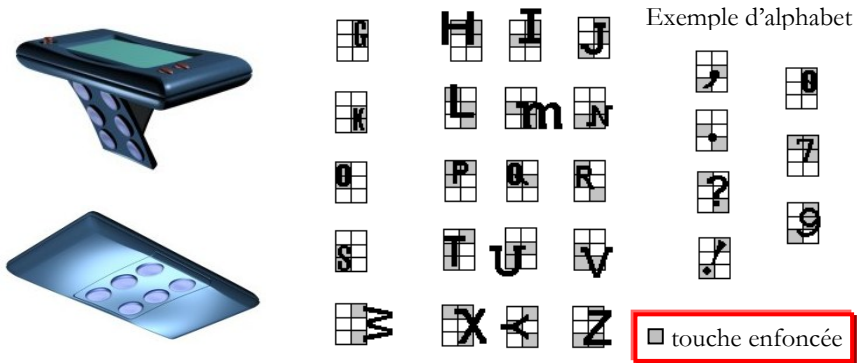


Figure 35. Clavier GKOS à 6 boutons

Le clavier Twiddler est un clavier à accords qui peut être utilisé en mode clavier réduit multi-tapes. Il a comme particularité de donner un retour visuel sur les caractères accessibles (Lyons *et al.*, 2004b) (figure 36), ce qui apporte une aide à l'utilisateur pour trouver les touches à sélectionner en mode multi-tapes et en mode à accords. Dans ce dernier mode, un code de couleur est utilisé pour désigner les touches à sélectionner tout comme le clavier multi-tapes FrogPad. Ce type de clavier a pour but d'équiper des téléphones. Il est conçu pour une interaction avec une seule main. Une étude comparative des deux approches a montré qu'avec apprentissage le mode « accords » atteint un seuil plus performant que le mode multi-tapes (tableau 1). Une autre particularité de ce clavier est qu'il permet de définir les mots ou les ensembles des lettres les plus fréquents avec des accords (Lyons *et al.*, 2004a).



Figure 36. Clavier Twiddler

Pour aider les utilisateurs novices à manipuler des accords, des aides ont été apportés pendant la phase d'apprentissage (Lyons *et al.*, 2005) : apprentissage progressif des accords en partant d'accords simples vers des accords plus compliqués ainsi qu'un affichage virtuel du clavier avec mise en surbrillance des touches à sélectionner pour réaliser un accord. Des tests ont montré que ces aides améliorent les performances pendant les vingt premières minutes d'apprentissage. Par contre, pendant l'évaluation où ces aides sont retirées, les performances ne diffèrent pas des utilisateurs n'ayant pas eu ces aides pendant leur apprentissage. Le but de ces aides est surtout d'attirer de nouveaux utilisateurs vers la saisie par accords.

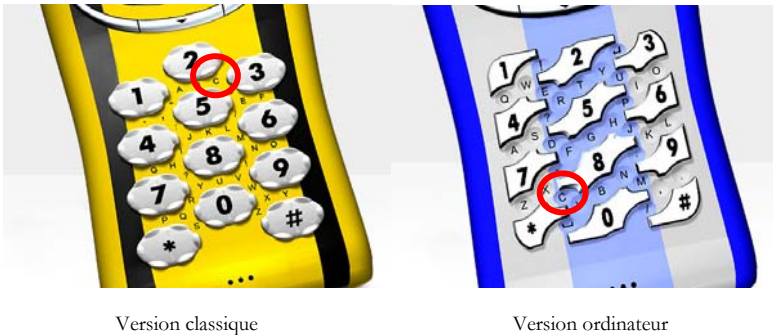


Figure 37. Claviers Q12

Le clavier Q12 (Softava Company, 2000) est un clavier destiné aux téléphones mobiles. Il est basé sur les douze touches. Les caractères sont dessinés entre les touches. C'est un clavier à accords, car pour atteindre une lettre il suffit d'appuyer simultanément les touches avoisinantes et l'ordre n'a pas d'importance. Pour lever toute ambiguïté sur les touches avoisinantes, leur forme est conçue de sorte que l'on ne puisse appuyer sur les autres touches. L'avantage de ce type de clavier est sa simplicité puisqu'il n'est pas besoin d'apprendre les accords. Sur la figure 37, deux modèles de répartition des lettres sont proposées ainsi que des formes différentes de boutons. À gauche, la version « classique », les lettres sont réparties dans l'ordre alphabétique : la lettre 'c' est obtenu par l'accord 2-3-5. À droite la version « ordinateur », les lettres sont réparties comme sur un clavier Qwerty : la lettre 'c' est obtenu par l'accord *-8.

2.4 Claviers configurables

Une nouvelle génération de claviers est en train de naître avec le clavier Optimus dont la première version noir et blanc est commercialisée depuis mai 2007 (Lebedev, 2005). Chacune de ses touches est dotée d'un mini écran de type OLED leur permettant d'être entièrement configurables. Ainsi le clavier peut selon le contexte être de type AZERTY ou personnalisé en fonction de l'application ou du souhait de l'utilisateur (figure 38). Son coût reste pour le moment élevé.

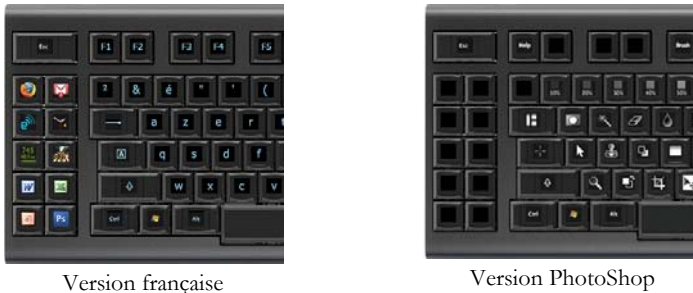


Figure 38. Clavier Optimus

Le clavier DX1 Input system de Ergodex (Case, 2004) a été pensé pour les joueurs afin de leur offrir une liberté totale d'action sur le contrôle des jeux. Le clavier est une surface plane sur laquelle l'utilisateur va pouvoir placer au maximum 25 touches reliées au clavier par une technologie sans fil (figure 39). La flexibilité est double :

- l'utilisateur peut attribuer les fonctions qu'il désire par rapport au logiciel qu'il veut piloter. Les fonctions peuvent être simples ou complexe (par exemple, représentant une chaîne de commandes)
- il peut disposer les touches comme il le souhaite sur le clavier et les déplacer à tout moment. La surface du clavier peut aussi être paramétrée, ce qui permet d'apporter des annotations autour des touches.



Figure 39. Clavier DX1 Input system

Il existe également des claviers logiciels qui cherchent à proposer une grande flexibilité afin de fournir une ergonomie adaptée à l'utilisateur, au contexte, au besoin, etc. Pour cela ces claviers fournissent des dispositions alternatives, des dispositions différentes d'un clavier standard, non systématiquement optimales en terme de placement des touches ou de vitesse de saisie, mais adaptées à un usage ou un utilisateur particulier. Des claviers logiciels (Ballayer, 2005) comme WiVik©, Click'N'Type© et Clavicom© proposent, outre les configurations standard pour un grand nombre de pays différents, de nombreuses dispositions alternatives. Ainsi la

première disposition de la figure 40 présente les touches du clavier WiVic© selon un ordre de fréquence d'apparition dans le texte (à gauche pour les plus fréquentes). Cette disposition s'adapte bien à une sélection par défilement alors que la seconde disposition de la figure 40 (avec la touche espace au centre) est mieux adaptée pour une sélection par pointeur.



Figure 40. Deux dispositions alternatives de WiVic©

En dehors des deux dispositions précédentes, le clavier WiVic© propose un nombre conséquent de dispositions alternatives. Une disposition particulière, appelée « mode Quadrant », est susceptible de convenir aux personnes dont les gestes manquent de précision (figure 41). Il s'agit ici d'un clavier réduit multi-tapes.

La configuration des claviers Click'N'Type©, Clavicom© et WiVic© est réalisée via un fichier éditable avec un simple éditeur de texte et qui sera chargé au démarrage. Un langage de programmation permet par exemple la définition des touches et de leur placement, des animations visuelles et sonores, des changements dynamiques de la disposition des touches, des macros, etc. La complexité de ce langage rend difficile l'accès à ce fichier de configuration par des non initiés.

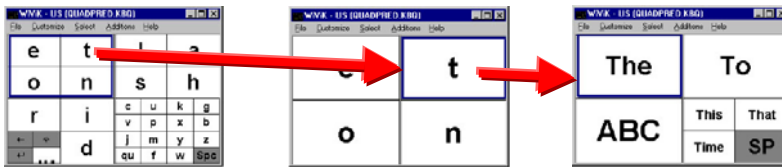


Figure 41. WiVic© en mode « Quadrant »

Destiné aux utilisateurs finaux, le logiciel CNTDesigner (Polital, 2003) propose de définir une configuration pour Click'N'Type©, selon un mode WYSIWYG (figure 42). Cependant, il ne permet pas d'aller au-delà des fonctionnalités de Click'N'Type©. Il est par exemple impossible de définir une interaction comme le mode « Quadrant » de WiVic©:

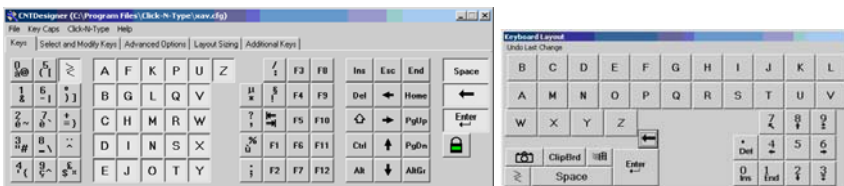


Figure 42. CNTDesigner

Enfin, l'environnement de conception de clavier logiciel SOKEYTO (Vella *et al.*, 2004) propose une interface de conception WYSIWYG, centrée utilisateur,

permettant de concevoir et de tester des claviers logiciels pour répondre aux besoins spécifiques des utilisateurs, en particulier différentes formes de handicaps moteurs (figure 43). Pour y parvenir, SOKEYTO intègre des modèles neurophysiologiques théoriques tels que la loi de Fitts, qui facilitent le calcul des performances théoriques de la disposition.

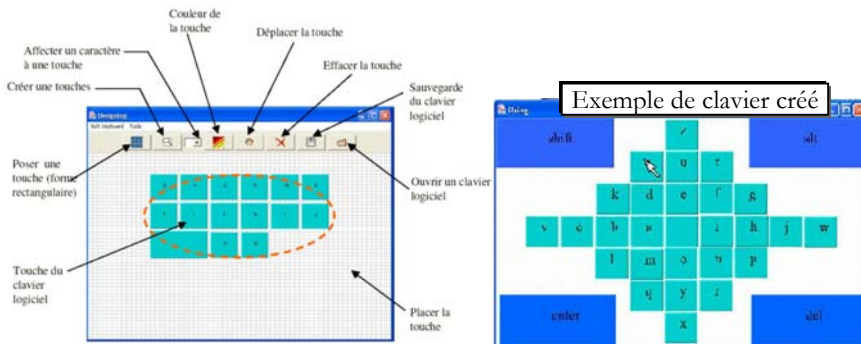


Figure 43. Interface de conception SOKEYTO

2.5 Performances des claviers à touches

Le tableau 1 donne les performances des différents claviers à touches présentés précédemment. Seules deux types de clavier n'apparaissent pas dans le tableau :

- les claviers configurables (cf. 2.4), car leurs performances dépendent de la configuration spatiale des touches choisie par l'utilisateur.
- les claviers dédiés aux personnes handicapées car leurs performances ne sont pas comparables, les utilisateurs étant très différents selon les études. Par exemple, pour des tests avec WiVik les utilisateurs atteignent à peine une vitesse de 1 wpm.

Les vitesses indiquées dans le tableau tiennent compte du niveau des utilisateurs. La vitesse minimale correspond au novice et la vitesse maximale à l'utilisateur expert. L'expert est parfois considéré par certains auteurs comme un utilisateur ayant quelques heures ou quelques sessions de pratique seulement. Les heures de pratique mentionnées dans le tableau ne sont par forcément consécutives. L'écart entre expert et novice peut être faible, mais il peut permettre de montrer la capacité d'apprentissage. Il est également difficile de préciser le nombre de doigts utilisés selon le type de clavier car les protocoles de tests qui ont permis d'obtenir les performances ne le mentionnent pas toujours. De plus, les claviers présentés ne sont pas nécessairement exclusifs à une prise en main. Par exemple, les claviers réduits de type téléphone mobile peuvent être pris dans une main et être utilisés avec un ou plusieurs doigts de l'autre main ou bien être pris à deux mains et être utilisés avec le pouce de chaque doigt. Enfin, les conditions de test sont parfois très différentes. Par exemple, certains tests ont été réalisés en situation « tête levée » que l'on peut rapprocher d'une situation de mobilité extrême alors que d'autres tests ont été réalisés en situation de bureau.

Type	Clavier	Vitesse (wpm)	
		Novice	Expert
Claviers standard	Qwerty	28 au début de l'expérience (MacKenzie et Zhang, 1999) 8,9 théorique (MacKenzie <i>et al.</i> , 1999)	45 après 15h (MacKenzie et Zhang, 1999)
	Qwerty logiciel	20,2 après 1 phrase sans apprentissage (MacKenzie <i>et al.</i> , 1999)	22,9 après 1 ou 2h (MacKenzie <i>et al.</i> , 1999) 25 après 10h (Isokoski, 2004)
	Azerty logiciel	inconnu	14,2 après quelques heures (Vella et Vigouroux, 2004)
	Canesta à projection	46,6 après 2mn (Roeber <i>et al.</i> , 2003)	inconnu
Claviers optimisés par agencement	Dvorak	8,5 après 1 phrase sans apprentissage (MacKenzie <i>et al.</i> , 1999)	38,7 théorique (MacKenzie <i>et al.</i> , 1999)
	XPeRT	20 après 2h (XPeRT, 2003)	40 après 20h (XPeRT, 2003)
	OPTI	17 (MacKenzie et Zhang, 1999)	40 après 15h (MacKenzie et Zhang, 1999)
	OPTI-II	inconnu	42,37 théorique (Soukoreff, 2002) 42,82 théorique (Raynal et Vigouroux, 2005b)
	FITALY	8,2 après 1 phrase sans apprentissage (MacKenzie <i>et al.</i> , 1999)	36 théorique (Zhai <i>et al.</i> , 2002) 41,96 théorique (Soukoreff, 2002) 55,9 théorique (MacKenzie <i>et al.</i> , 1999) 41,96 théorique (Soukoreff, 2002) 40,96 théorique (Raynal et Vigouroux, 2005b)
	Metropolis I	inconnu	42,15 théorique (Soukoreff, 2002)
	Metropolis II	inconnu	42,94 théorique (Soukoreff, 2002)
	Atomik	inconnu	41 théorique (Zhai <i>et al.</i> , 2002) 45 théorique (Zhai et Kristensson, 2003)
	GAG I	inconnu	46,43 théorique (Raynal et Vigouroux, 2005b)
	FOCL	5,8 (Bellman et Mackenzie, 1998)	10 après 2h (Bellman et Mackenzie, 1998)
Claviers optimisés par outils	Fisheye	25% plus rapide, 25% d'erreurs en moins et distance mot entré/mot attendu réduite de 15% par rapport à un azerty logiciel (Raynal et Truillet, 2007)	inconnu
	Qwerty avec menu radial		25 après 10h (Isokoski, 2004)
	KeyGlass	inconnu	inconnu
	Indices visuels	9,37 (Magnien <i>et al.</i> , 2004) (Clavier sur assistant personnel)	inconnu
	POBox	inconnu	inconnu
	Clavier bi-manuel à deux joysticks	7,08 ±0.46 (Wilson et Agrawala, 2006) 6,75 après 40 mn de pratique (Sandnes et Aubert, 2007)	inconnu
Claviers réduits multi-tapes	T12	8,1 après 1 phrase sans apprentissage (MacKenzie <i>et al.</i> , 1999)	22,6 théorique (MacKenzie <i>et al.</i> , 1999)
	Nokia 3210	7,98 (James et Reischel, 2001)	7,93 (James et Reischel, 2001)
	T12 virtuel	9,6 après quelques heures (Vella et Vigouroux, 2004)	inconnu
	MultiPlay logiciel	10,5 (KeyTouch, 2003)	inconnu

	MessageEase	inconnu	30-37 théorique (Nesbat, 2003)
	MessageEase logiciel	inconnu	50-63 théorique (Nesbat, 2003)
	SpeedScript	inconnu	57.12 (SpeedScript, 1999)
	DotNote	inconnu	29,46 théorique (Mackenzie et Soukoreff, 2002c)
	FrogPad	inconnu	40 (FrogPad, 2006)
	Half-QWERTY	inconnu	Au moins 40 après 10h (Matias <i>et al.</i> , 1996)
	In10Did	inconnu	inconnu
	Twiddler mode multi-tapes	8,2 (Lyons <i>et al.</i> , 2004b)	19,8 après 7h (Lyons <i>et al.</i> , 2004b)
Claviers réduits mono-tape	Clavier logiciel réduit avec répartition aléatoire des caractères sur les touches	5,82 (Magnien <i>et al.</i> , 2004)	inconnu
	T9	9,09 (James et Reischel, 2001) 5 (Cockburn et Siresena, 2003)	20,36 (James et Reischel, 2001) 10,8 après 1h (Cockburn et Siresena, 2003)
	EQ3	18,9 théorique (Castellucci, 2007)	46,7 théorique (Castellucci, 2007)
	LetterWise	7,2 (MacKenzie <i>et al.</i> , 2001)	21 après 10h (MacKenzie <i>et al.</i> , 2001)
	TiltText	7,42 (Wigdor et Balakrishnan, 2003) au début du test	13,6 après 160mn (Wigdor et Balakrishnan, 2003)
	Claviature	inconnu	inconnu
	Fastap	7,1 (Cockburn et Siresena, 2003)	9,3 après 1h (Cockburn et Siresena, 2003)
	Mini-Qwerty (Targus et Dell)	31 (Clarkson <i>et al.</i> , 2005)	60 après 7h (Clarkson <i>et al.</i> , 2005)
Claviers réduits à accords	Clavier du système NLS	inconnu	inconnu
	CyKey Plus®	inconnu	inconnu
	GKOS	20 (Wobbrock, 2006)	40 (Wobbrock, 2006)
	Twiddler mode accords	4,3 (Lyons <i>et al.</i> , 2004b)	26,2 après 7h (Lyons <i>et al.</i> , 2004b) 47 après 25h (Lyons <i>et al.</i> , 2004a)
	Q12	inconnu	inconnu

Tableau 1. Performance des claviers à touches

Tous les claviers logiciels fournissent le même type de retour visuel : le caractère codé par une touche est indiqué directement sur la touche. On est face à un retour visuel direct. Le mode d'interaction reste aussi le même pour l'ensemble des claviers logiciels. Leurs performances sont des maximums théoriques pour des utilisateurs experts. Selon les méthodes utilisées par (Zhai *et al.*, 2002) ou (Soukoreff, 2002) pour le calcul, celles-ci varient quelque peu comme on peut le voir dans le tableau 1. (Zhai *et al.*, 2002) se basent sur les statistiques de digrammes de Mayzner et Tresselt alors que (Soukoreff, 2002) utilise le modèle décrit dans (Soukoreff et MacKenzie, 1995) avec quelques améliorations.

Concernant le clavier Qwerty, la valeur 28 wpm représente des performances au début d'une expérience. Cela ne reflète pas forcément les performances de personnes n'ayant jamais manipulé un tel clavier. Il est très difficile voire impossible

de trouver des utilisateurs n'ayant jamais utilisé un tel clavier. Aussi la valeur théorique est certainement plus représentative.

Si un grand nombre de personnes sait à présent se servir avec plus ou moins d'habileté d'un clavier T12, certaines personnes ont beaucoup de mal à utiliser le système de prédiction T9. Cela peut expliquer que les performances minimales soient équivalentes. Pour Multiplay, la valeur novice correspond à un test simple de répétition (10 fois) de la phrase « Quick brown fox jumps over the lazy dog. » sur un simulateur logiciel. Les performances du clavier FrogPad® sont également à relativiser, car les chiffres annoncés (40 wpm) pour le mode expert correspondent selon le fabricant à dix heures de pratique, ce qui semble difficile à atteindre selon (Carey, 2004). Il est également à remarquer que la petite taille du clavier Fastap par rapport au nombre de touches donne des gains de performances minimes, même si le confort de saisie apparaît meilleur.

Hormis les claviers CyKey® et GKOS®, tous les claviers offrent un retour visuel direct sur la touche ou sur une représentation de la touche sur l'écran. Ces claviers sont donc logiquement handicapés dans leurs performances vis-à-vis d'un utilisateur débutant, celui-ci ne sachant pas quelle combinaison de touches code quel caractère. Pour le clavier GKOS, les chiffres indiqués pour le débutant correspondent d'ailleurs à un utilisateur qui a un peu de pratique. Les claviers à accords disposent cependant d'une grande capacité de progression, ce qui compense leurs faibles performances au départ.

Toutefois, dans un contexte standard d'utilisation, les claviers Qwerty/Azerty restent majoritairement utilisés. Deux raisons évidentes peuvent justifier cette tendance : le temps d'apprentissage et les habitudes des utilisateurs sont jugés trop grandes pour changer. La capacité de progression sur les claviers standard est conséquente, parce que le temps d'apprentissage est long. Pour des typographes professionnels, 80 wpm sur claviers Qwerty est courant. La différence entre Qwerty/Azerty et Dvorak au niveau de la vitesse n'est pas suffisamment importante (4%) pour justifier un nouvel apprentissage. Par contre, le gain se fait probablement au niveau du confort.

Chaque clavier présenté dans cette table possède ses avantages et ses inconvénients. Par exemple, le clavier mini-Qwerty n'est pas idéal en mode aveugle, mais ne nécessite pas d'apprentissage alors que le clavier Twidler est plus performant en mode aveugle que le clavier mini-Qwerty, mais demande un apprentissage conséquent pour connaître les accords (Clawson *et al.*, 2005). Il revient à chaque individu d'évaluer la situation en fonction des caractéristiques de chaque clavier décrites dans cette partie.

3 Claviers gestuels

Un clavier gestuel est un système logiciel de saisie de texte tel que l'interaction est réalisée par des gestes que le système interprète en tant que caractère ou commande. Les claviers à gestes apportent une tentative de solution au problème de saisie de texte sur périphériques informatiques réduits (Isokoski, 1999). Ils sont souvent associés à un contexte d'utilisation nomade et à un dispositif matériel dédié, comme un touchpad ou une surface tactile munie d'un stylet. De ce fait, ils sont nombreux sur les assistants numériques. Il est vrai que la souris n'est pas la plus appropriée ici, autant par son manque de précision que par son inadaptation aux assistants numériques. Parmi l'existant, le geste nécessaire à l'interaction peut avoir trois natures complètement différentes :

- écriture manuscrite standard.

- geste symbolique.
- geste vers une cible.

3.1 Claviers gestuels à reconnaissance d'écriture

La reconnaissance d'écriture est une approche ancienne qui consiste à écrire naturellement les caractères sans contrainte. Généralement cela se fait avec un stylet ou un doigt sur une surface tactile par analogie à l'écriture manuscrite « standard ». La reconnaissance se fait soit à partir de l'information statique résultant de l'action de l'écriture, soit à partir de l'information dynamique qui est l'action d'écriture.

L'écriture n'est généralement pas basée sur un alphabet pré-établi, l'utilisateur reste libre dans ses gestes. Toutefois, une phase d'apprentissage du système s'avère souvent nécessaire afin de minimiser les erreurs de reconnaissance. De plus, même s'il est possible parfois d'écrire en continu, les systèmes de reconnaissance demandent le plus souvent une écriture individuelle des caractères.

Les claviers à reconnaissance d'écriture sont immédiatement utilisables car l'écriture est intuitive et ne demande pas l'apprentissage d'un nouvel alphabet. Cependant cette technique reste lente comparée aux techniques de frappes classiques présentées dans la partie précédente. De plus les difficultés de reconnaissance restent d'actualité et imposent souvent l'usage de plusieurs modes et/ou de corriger fréquemment les erreurs. Nous ne détaillerons pas plus cette catégorie de clavier car cela représente un champ de recherche à part entière. Le lecteur intéressé pourra toujours se reporter sur des articles état de l'art tels que (Plamondon et Srihari, 2000) et (Bunke, 2003).

3.2 Claviers gestuels symboliques

Un clavier à gestes symboliques implémente une reconnaissance du geste pour l'associer à un unique caractère. On définit un geste comme une succession de tracés élémentaires : le geste reste ininterrompu. La combinaison des tracés élémentaires réalisés détermine le caractère saisi. La reconnaissance des caractères manuscrits posant encore de nombreux problèmes, les claviers à gestes symboliques introduisent un alphabet plus simple, et établissent la correspondance entre un symbole de cet alphabet et un caractère de l'alphabet latin (ou la commande souhaitée). Ainsi, l'utilisateur dessine le symbole et c'est la lettre correspondante qui apparaît à l'écran. C'est généralement le seul retour visuel dont l'utilisateur dispose pour l'ensemble des claviers présentés dans cette partie et il est important de noter que le confort et les performances atteignent un niveau élevé seulement si on peut se passer de vérification lors de la saisie (saisie « à l'aveugle » ou encore « tête levée »).

Historiquement le premier des claviers à gestes symboliques, Unistroke (Goldberg et Richardson, 1993) définit un alphabet dont chacun des symboles s'obtient par un et un seul tracé élémentaire. L'alphabet a été défini en fonction de trois critères : simplicité d'apprentissage, différenciation maximale entre deux symboles et rapidité de la saisie. L'écart entre deux symboles est pensé pour faciliter l'implémentation et résoudre l'ambiguïté entre deux caractères proches dans leur forme latine ('f' et 't', 'c' et 'e' par exemple). Les voyelles, lettres les plus courantes, sont associées à un symbole simple pour accélérer la vitesse de saisie (figure 44).

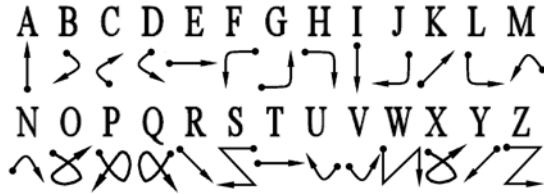


Figure 44. *Alphabet Unistroke*

Avec un seul tracé élémentaire par caractère, Unistroke a visé la simplicité, et a été conçu pour permettre une saisie « à l'aveugle ». Mais le concept a été introduit trop tôt ou a effrayé les utilisateurs potentiels, car aucun PDA n'en fait usage de nos jours. Cependant, l'alphabet et les concepts introduits servent toujours de référence pour la création d'un clavier à gestes symboliques (Isokoski, 2001). Les successeurs d'Unistroke préfèrent malgré tout mettre en avant d'autres avantages.

Graffiti est le clavier à gestes symboliques développé par Palm Computing pour sa gamme d'assistants numériques (Fleetwood *et al.*, 2002). Comme avec Unistroke, la plupart des symboles sont dessinés en un seul tracé élémentaire. Mais de manière stricte Graffiti ne respecte pas la philosophie de son prédécesseur : certains caractères sont obtenus par deux tracés élémentaires (tel le 'K' sur la figure 45) et les voyelles ne sont pas les symboles les plus rapidement tracés. Le but recherché est ici de se rapprocher au maximum des caractères latins pour permettre un apprentissage plus rapide, même si cela se fait aux dépens de la simplicité des tracés.

La figure 45 indique que l'apprentissage peut être perturbé car certains symboles s'appuient sur des minuscules (par exemple 'Y') et d'autres sur des majuscules (par exemple 'J'). Certains symboles, comme le 'Y', ressemblent à s'y méprendre à des caractères latins : on peut penser que l'algorithme de reconnaissance est moins précis et surtout moins rapide que celui d'Unistroke. Cette similarité avec l'alphabet latin pose des problèmes d'ambiguïté pour la reconnaissance. Par exemple, le système confond un 'H' avec un 'L' si la barre horizontale du 'L' est trop incurvée. Les concepteurs de Graffiti ont profité de l'avantage qu'ils avaient de contrôler matériel et logiciel pour résoudre matériellement une partie des ambiguïtés dues à leur alphabet. Ils ont introduit deux zones de saisie distinctes, l'une pour les lettres et l'autre pour les chiffres. Mais cela ne résout pas tout, et le système pose parfois des difficultés si on ne lève pas les ambiguïtés restantes « manuellement ». En pratique, celui-ci distingue d'ailleurs encore avec difficulté un 'h' d'un 'l' ou même d'un 'n'.

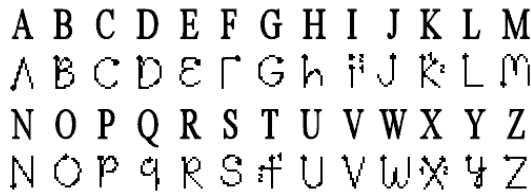


Figure 45. *Alphabet Graffiti2*

Graffiti s'est imposé en tant que norme sur Palm. Le fait qu'il soit installé par défaut sur les PDA sous PalmOS n'y est pas étranger et, bien que Graffiti ne soit

pas parfait, il fonctionne suffisamment bien pour que les utilisateurs plébiscitent sa facilité d'apprentissage. Pourtant les performances pures peuvent décevoir et la saisie « à l'aveugle » est difficile car il faut rester dans la bonne zone de saisie.

Le clavier MDITIM est un clavier à gestes symboliques défini par (Isokoski, 1999) pour s'abstraire du périphérique d'entrée et ainsi proposer un clavier générique pour l'ensemble des systèmes informatiques, mobiles en particulier. En permettant une quasi-universalité, MDITIM va à l'encontre des autres claviers à gestes symboliques, qui proposent un système de saisie associant logiciel avec matériel dédié (Fleetwood *et al.*, 2002), (Smith, 2000). Afin d'autoriser la plus grande généricité possible, le clavier utilise le minimum de fonctionnalités communes à l'ensemble des périphériques d'entrée : quatre directions ou touches pour indiquer quatre directions Nord, Sud, Est et Ouest. Ceci permet par exemple la saisie à l'aide d'une souris, d'un joystick, ou plus simplement de quatre touches sur un clavier physique. Il s'adapte même très bien à une saisie à l'aide de mouvements par inclinaison sans aucun tracé écran (« tilting keyboard »). L'alphabet défini est visible en figure 46.

Pour tracer un caractère on se déplace en partant d'un point sur un et un seul axe Nord=<=>Sud ou Est=<=>Ouest à la fois : les symboles de cet alphabet n'utilisent pas de déplacements en diagonale car ils ne sont pas accessibles sur tous les périphériques d'entrée. Cela simplifie la reconnaissance du symbole : on ne peut aller que dans trois directions bien précises : « à gauche », « à droite », ou « en arrière » (un déplacement « en avant », dans la même direction, correspond à l'annulation du caractère en cours de tracé).

MDITIM permet une saisie continue : il est inutile de valider le caractère entré pour saisir le suivant. Avec certains périphériques d'entrée, une saisie tête levée est possible. Ainsi il est adapté aux personnes déficientes visuelles ou dont les gestes manquent de précision. L'alphabet introduit est conforme à Unistroke mais MDITIM hérite aussi des défauts : un symbole codant un caractère est assez éloigné de l'équivalent latin de ce caractère. Certains symboles sont assez complexes à tracer avec un stylet, mais la saisie au joystick par exemple est aisée. Une étude pratique des performances de MDITIM réalisée par (Isokoski, 1999) a montré que ce clavier est plus lent que les systèmes existants. Comme une majorité de claviers à gestes symboliques, le retour visuel est inexistant. Malgré ces inconvénients, l'idée d'un alphabet universel mérite d'être approfondie car un tel alphabet risque de devenir, sinon incontournable, du moins indispensable.

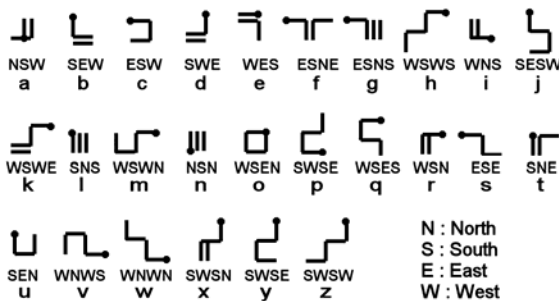


Figure 46. Alphabet MDITIM

Dédié aux assistants numériques, EdgeWrite est un clavier à gestes symboliques conçu pour permettre une grande tolérance aux imperfections de tracé. L'alphabet a été mis au point non pas en fonction des tracés élémentaires des

symboles, mais en fonction de l'ordre dans lequel on passe par les quatre coins d'un carré (figure 47 à gauche). Tous les caractères ressemblent plus ou moins à leurs homologues de l'alphabet latin : soit à la lettre minuscule soit à la lettre majuscule. Les gestes s'inscrivent dans un guide (figure 47 à droite), les possibilités d'erreurs sont ainsi réduites. Le guide prend la forme d'un carré de plastique de taille réduite, qui « arrête » le stylet et limite par conséquent l'amplitude des mouvements. L'alphabet obtenu est conforme à Unistroke.

Tout comme MDITIM, EdgeWrite offre une grande généricité car il utilise des fonctionnalités communes à de nombreux périphériques d'entrée : quatre entrées (coin) et une validation. Ceci a permis d'introduire de nombreuses versions en fonction du périphérique d'entrée (Wobbrock et Myers, 2005). EdgeWrite offre un nombre conséquent d'avantages (Wobbrock *et al.*, 2003). En premier lieu, la rapidité de la reconnaissance. Le système de saisie n'a pas un dessin à reconnaître, mais juste une liste de points à parcourir afin de déterminer le caractère. Alors que l'alphabet présente les différents symboles comme des gestes, ceux-ci sont traités comme une liste de points. L'algorithme de reconnaissance s'exécute au pire des cas en un temps linéaire. Afin de faciliter l'apprentissage de l'alphabet une aide intégrée a été proposée récemment (Martin et Isokoski, 2008). Cette aide a montré qu'elle pouvait remplacer l'aide papier sans perte de performance.

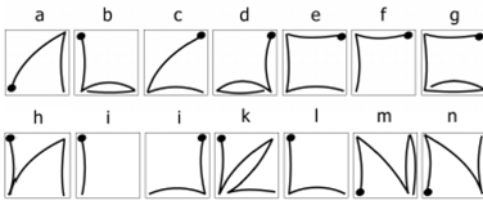


Figure 47. Alphabet EdgeWrite et son guide de saisie

La dernière version d'EdgeWrite s'est enrichie d'une complétion de mots appelée Fisch (Wobbrock *et al.*, 2006). Dès que possible, un mot est proposé dans chaque coin de la zone de saisie (Figure 48 à gauche). Pour valider un mot proposé, l'utilisateur doit indiquer la fin de la saisie de la lettre en cours par une boucle dans le dernier coin (Figure 48 au milieu). Le mot de ce coin est automatiquement sélectionné mais si l'utilisateur veut un autre mot, il lui suffit d'aller dans l'un des trois autres coins (Figure 48 à droite). La technique de validation peut toutefois dépendre du périphérique de saisie.



Figure 48. Complétion de mot intégrée dans EdgeWrite : geste pour 't', 'the' et 'they'

L'association du guide de saisie et la non prise en compte des tracés entre les coins dans la reconnaissance permettent de rendre ce système accessible aux personnes souffrant d'un handicap moteur. Si l'utilisateur n'utilise pas le retour

visuel de la complétion de mot, une saisie « à l'aveugle » s'avère possible tant que le pointeur reste dans le guide de plastique ou dans la zone de saisie.

(Chau *et al.*, 2006) a proposé une version dédiée aux téléphones portable en intégrant un joystick isométrique (figure 49). Cette adaptation d'EdgeWrite permettrait de répondre aux difficultés d'interaction en situation de mobilité. Deux versions sont proposées : l'une où le joystick est sur la face avant pour un contrôle avec le pouce et l'autre sur la face arrière pour un contrôle avec l'index. L'idée est de tester les performances en position assise et situation de mobilité. À l'heure actuelle, un simple test préliminaire avec deux des auteurs a été mené pour comparer les deux joysticks en situation fixe : il semble que l'utilisation avec le pouce soit plus efficace. L'utilisation d'un joystick isométrique offre à l'utilisateur un minimum d'espace pour la saisie : il fonctionne par pression du doigt et ne nécessite pas de déplacement du doigt. Il est plus précis qu'un joystick classique à quatre directions.

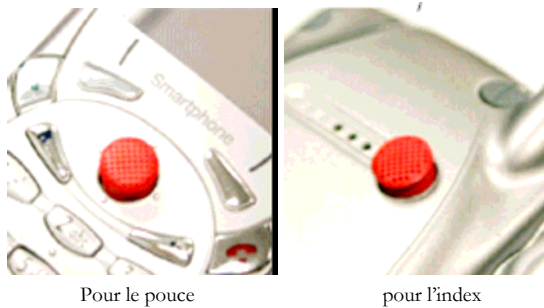


Figure 49. Saisie EdgeWrite avec joystick sur face avant ou face arrière.

ThumbScript implémente un principe similaire à celui d'EdgeWrite, à savoir considérer une suite de points plutôt que de tracés élémentaires, mais utilise une grille de 3x3 points au lieu de 2x2 (figure 50). Le geste associé à un caractère commence sur l'un des 8 points du bord de la grille et finit sur un autre point du bord de la grille : le tracé de chaque symbole de l'alphabet présenté ci-dessous passe systématiquement par la zone centrale. Thumbscript offre une généricité relative car il peut s'adapter à tout périphérique à 8 sorties pour les caractères plus 4 sorties spécifiques pour changer de mode.



Figure 50. Grille de saisie et alphabet ThumbScript.

L'alphabet introduit par ThumbScript est conforme à Unistroke et les symboles ne ressemblent pas aux lettres latines. L'avantage d'utiliser une grille de points permet de résoudre de nombreux problèmes d'ambiguïté rencontrés avec Graffiti et même Unistroke. Dans le cas d'une saisie par pointeur, la grille de points

s'affiche au moment de la saisie, servant de guide visuel. Cette grille étant fixe, les tracés doivent toujours être réalisés au même endroit, et surtout être d'une longueur déterminée, à un pourcentage d'erreur près. En l'absence de cadre matériel, il faut conserver le regard sur la grille pour contrôler la trajectoire du pointeur. De telles conditions ne sont ni confortables ni propices à des performances élevées.

La société a introduit un périphérique dédié de la taille d'un étui à cartes, le « ThumbScript Feather Touch », permettant une saisie à un doigt (figure 51 à gauche) Ce périphérique vient donc pallier le défaut du logiciel, et permettrait d'être jusqu'à 25% plus rapide avec ThumbScript qu'avec Graffiti (Thumbscript Development, 2000). Puisqu'il nécessite un matériel dédié, on comprendra bien que ce système de saisie est plus destiné à être utilisé comme système par défaut plutôt que comme système de saisie alternatif. C'est pourquoi la société dispose également d'un prototype intégré de démonstration (figure 51 à droite) (Smith, 2000).

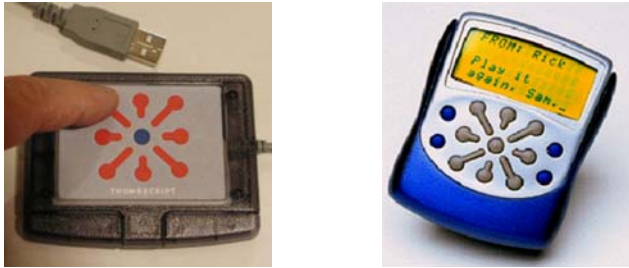


Figure 51. *ThumbScript Feather Touch et ThumbScript pager.*

On peut remarquer la grande ressemblance du clavier avec un clavier de type téléphone mobile. Utilisé sur un téléphone portable, ThumbScript serait certainement très efficace et offrirait l'avantage de pouvoir réduire encore la taille du clavier et des touches : il ne serait plus nécessaire d'y indiquer les symboles associés à chaque touche, étant donné que ce ne serait plus la touche qui coderait un symbole mais un tracé élémentaire. De tels systèmes de saisie sont bien adaptés pour les appareils portatifs personnels car leur confort d'utilisation s'améliore au fur et à mesure que l'utilisateur en acquiert la maîtrise. Mais ils sont à éviter pour les appareils d'usage public (comme une borne à écran tactile) car l'absence quasi-totale de retour visuel ne permettra pas au novice de s'en servir aisément.

Symbol Creator (Miniotas et al., 2003) et Glyph (Poirier, 2005) présentent une approche originale afin de simplifier la reconnaissance de l'alphabet latin. Les caractères sont construits à l'aide d'une séquence de deux ou trois primitives visibles en figure 52. Alors que Symbol Creator se limite aux lettres de l'alphabet, les primitives choisies pour Glyph permettent l'écriture de plus de lettres sans perdre l'analogie de la forme (les chiffres par exemple).



Figure 52. *Les primitives de Symbol Creator et de Glyph.*

La séquence de primitives à entrer est liée à l'analogie entre la forme des primitives et la forme du caractère latin désiré. La figure 53 montre des exemples de séquences pour les deux propositions. Il est à noter que si un caractère est obtenu

en sélectionnant les primitives liées à sa forme latine, les tests menés sur ces deux claviers se sont faits généralement avec des interactions sur des touches. Toutefois de part la nature gestuelle de leur conception, nous avons choisi de les ranger dans la catégorie claviers gestuels.

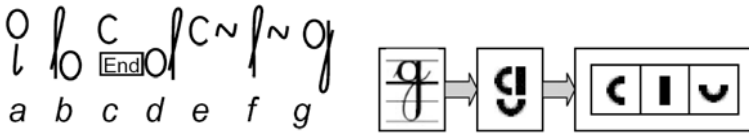


Figure 53. Exemples de séquences pour Symbol Creator et Glyph.

Alors que Glyph décompose un caractère en une séquence de 1, 2 ou 3 primitives et impose donc une validation, Glyph 2 utilise un alphabet étendu et permet de saisir n'importe quel caractère avec exactement deux primitives (Poirier et Belatar, 2006b). Cela permet d'éviter l'ajout de cette validation et uniformise l'alphabet. Une version nomade est présentée dans (Belatar, 2005). Une dernière version, UniGlyph, va plus loin encore en ne demandant l'usage que d'une primitive par caractère (Poirier et Belatar, 2006a) (Poirier et Belatar, 2007). Cette pratique induit inévitablement des ambiguïtés car le nombre de primitives est limité à 6. En se basant sur le même principe que le système T9® de Tegic Communications, UniGlyph utilise un système de prédiction pour lever ces ambiguïtés.

Symbol Creator (Miniotos *et al.*, 2003) et Glyph (Poirier, 2005) offrent une bonne généricité car ils demandent des périphériques dont le nombre de sorties est au moins égal au nombre de primitives plus éventuellement une validation. La saisie peut donc se faire par stylet ou clavier. La saisie s'apparente à une saisie consécutive en sélectionnant plusieurs touches (Wigdor et Balarishnan, 2004). Par nature, la vitesse de saisie est donc assez lente sans l'emploi d'une prédiction.

3.3 Claviers gestuels vers une cible

On introduit le terme de claviers gestuels vers une cible pour désigner un clavier à gestes, dont chaque geste correspond à un déplacement depuis la position courante du pointeur vers une cible généralement associée à un caractère. L'ensemble des caractères est généralement mis à disposition de l'utilisateur mais dans le cas d'un affichage incomplet, les caractères seront répartis sur plusieurs « pages ». La plupart de ces claviers se réinitialisent à leur état de départ lorsque le caractère est sélectionné. Pour un caractère donné, le geste de sélection est donc invariant. Dans certains cas, l'utilisateur qui maîtrise le système correctement peut désactiver le retour visuel, ce qui lui permet de tracer directement les gestes d'où une certaine ambiguïté entre ce type de claviers et les claviers à gestes symboliques présentés dans la suite. Il existe trois catégories de claviers gestuels vers une cible selon que l'on peut écrire, en un seul geste, un caractère, un mot ou une phrase.

Claviers gestuels vers une cible niveau caractère

Les claviers gestuels vers une cible à sélection discontinue sont tels que la saisie doit être interrompue (ou validée) après chaque caractère ou syllabe saisi. T-Cube, défini par (Venolia et Neiberg, 1994) se présente comme un ensemble de menus en anneaux (Hopkins, 1991). Les menus en anneaux permettent d'afficher un menu circulaire au dessus de l'espace de travail d'une fenêtre, plutôt qu'en ligne en haut de celle-ci. Initialement, T-Cube propose un menu comportant neuf possibilités : une zone centrale et huit zones périphériques (figure 54).

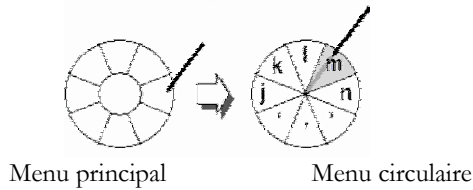


Figure 54. Menus avec T-Cube

Le système fonctionne de deux manières complémentaires. Pour l'utilisateur expert, il est possible de désactiver le retour visuel, afin de tracer directement les gestes correspondant aux parcours jusqu'au caractère souhaité. Pour le novice, un guidage visuel est proposé (« training mode »). Lorsque l'utilisateur maintient son pointeur pendant plus d'un tiers de seconde dans l'une des zones du menu principal (figure 54), un menu circulaire apparaît dont le contenu dépend de la zone sélectionnée. Afin qu'il ne soit pas caché par la main de l'utilisateur, ce menu est affiché avec un léger décalage par rapport au menu initial. Le sous-menu affiché, l'utilisateur peut sélectionner le caractère qu'il souhaite saisir grâce à un geste directif par rapport au centre du menu. Le caractère sélectionné est mis en évidence par un changement de couleur. Si l'utilisateur relâche le pointeur, le caractère est validé. Sinon, il est possible de tracer un geste dans une autre direction pour sélectionner un autre caractère. L'alphabet proposé est présenté par la figure 55. T-Cube utilise un seul niveau de 9 sous-menus comportant chacun 8 choix. On a donc un ensemble de 72 choix possibles, dont certains multiples accessibles par des touches de contrôle.

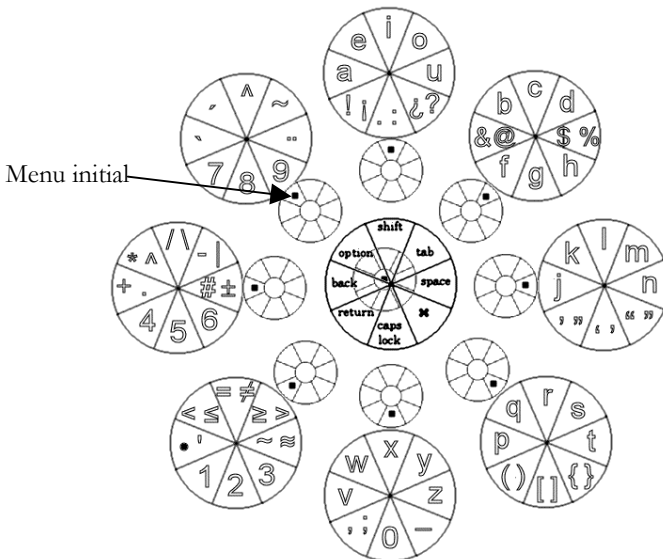


Figure 55. Alphabet T-Cube

En termes de nombre de gestes par caractères, T-Cube est très proche du minimum possible, on peut difficilement faire mieux avec un geste par caractère. Cependant, l'usage de ce clavier est d'autant plus minutieux que le menu circulaire est réduit, et à l'inverse d'autant plus lent qu'il est grand (à cause des déplacements plus amples du pointeur). Il est possible d'augmenter le nombre de niveaux et/ou

de choix par niveaux du menu pour avoir plus de caractères disponibles, mais les études réalisées par (Hopkins, 1991) ont montré que le temps de réponse de l'utilisateur croissait de manière linéaire avec le nombre de niveaux ; et que le taux d'erreur augmentait dramatiquement en passant de quatre choix par menus à huit ou douze quand la profondeur est supérieure à un niveau (Isokoski, 1999).

L'apprentissage du clavier se fait comme celui d'un clavier standard : le novice va aller chercher le caractère souhaité dans l'ensemble des sous-menus, alors que l'expert saura de mémoire où le trouver. Comme il existe un délai avant l'affichage, l'utilisateur expert a la possibilité de tracer directement le geste sans attendre le retour visuel. On se rapproche alors d'un clavier à gestes symboliques. T-Cube souffre cependant d'une limitation : la saisie requiert l'attention visuelle de l'utilisateur. Comme le menu initial est fixe, celui-ci est obligé de vérifier qu'il sélectionne bien la bonne zone de départ du geste. Ainsi, même pour un habitué, le contrôle visuel est obligatoire. D'où un confort de saisie relatif et des performances qui, si elles sont aussi bonnes que l'écriture manuelle, ne sont pas exceptionnelles. Malgré tout, T-Cube offre une alternative intéressante car il occupe moins d'un pouce carré à l'écran, tout en offrant un retour visuel suffisant.

Le VirHKey introduit une interaction à plusieurs niveaux (Martin, 2005a et 2005b). Il s'appuie sur un pavage hyperbolique, la pentagrigille, qui est tapissé de caractères comme indiqué sur la figure 56. L'agencement des caractères a été choisi pour son côté prédictible : les voyelles sont réparties sur le premier niveau dans le sens antihoraire. Puis à partir de chaque voyelle, les trois consonnes qui la suivent dans l'ordre lexicographique sont alors réparties dans le deuxième niveau toujours dans le sens antihoraire. S'il « reste » encore des consonnes, elles sont placées à partir de la troisième consonne dans l'ordre lexicographique dans le troisième niveau toujours dans le sens antihoraire. L'accès se fait alors hiérarchiquement, pour atteindre une lettre il faut passer pas ses « parents ». Par exemple pour accéder à la lettre Q, il conviendra de passer par la lettre O.

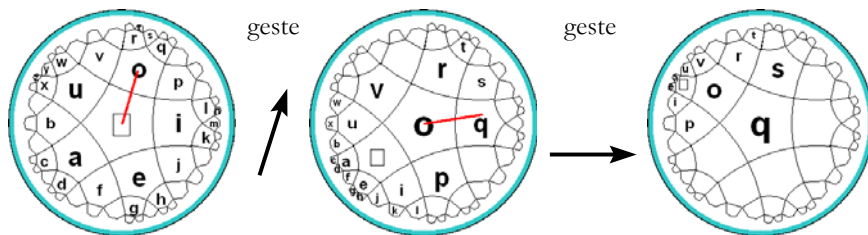


Figure 56. Un exemple d'interaction avec le VirHKey : le caractère q

A l'aide d'un stylet utilisé sur une surface sensitive, l'utilisateur interagit avec le clavier grâce à des gestes élémentaires, chiquenaudes, dirigées vers l'un des caractères du premier niveau. Il s'agit donc pour l'utilisateur d'indiquer un cadran parmi cinq possibles ; la précision demandée est donc de $2\pi/5$ au contraire des $2\pi/8$ requis pour T-Cube qui possède huit cadrans par sous-menu. Lorsque la longueur du geste atteint un seuil fixé, le caractère indiqué est amené au centre du pavage. Les autres caractères se déplacent en conséquence. L'utilisateur continue sans lever le stylet jusqu'à avoir amené au centre le caractère voulu. Le fait de lever le stylet entraîne la validation du caractère au centre du pavage et un retour du clavier à son état initial. La figure 56 montre une interaction pour la sélection du caractère q ; le geste requis est \nearrow . Tout au long des interactions, un retour visuel de l'état du clavier est disponible ainsi qu'un retour visuel des interactions de

l'utilisateur. Un retour sonore indique également à chaque fois qu'un geste élémentaire est validé.

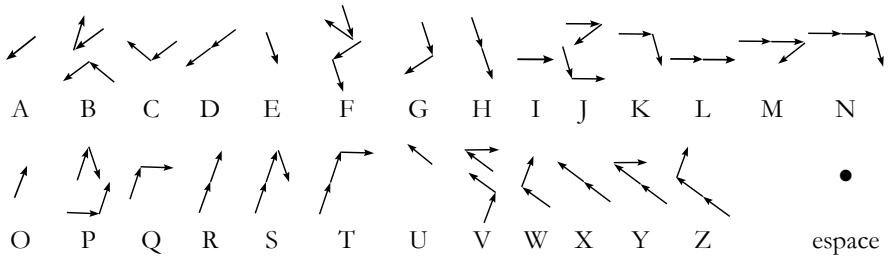


Figure 57. *Alphabet standard VirHKey*

La figure 57 présente l'alphabet ainsi obtenu. Il est conforme à Unistroke garantissant ainsi de bonnes performances. A la différence de T-Cube, un expert devrait pouvoir se passer totalement du retour visuel car le point de départ du geste n'a pas d'influence sur le caractère obtenu. Le retour visuel n'est nécessaire que dans les phases d'apprentissage. Cette approche offre également la possibilité d'interagir grâce à une interface de type clavier physique ou clavier logiciel réduits : cinq touches permettent la navigation plus une touche de validation. Dans ce cas, le VirHKey se rapproche d'un clavier à touches réduit à accords en séquence ; on accède à un caractère par appuis successifs sur une ou plusieurs touches (dépend de la méthode). Par nature, la vitesse de saisie est donc assez lente même si les tests montrent de bonnes performances.

Unigesture (Sazawal *et al.*, 2002) présente une interaction simplifiée à seulement un niveau. Les caractères sont organisés en 8 zones périphériques, 7 pour les lettres (3 à 4 lettres par zone) et une zone pour l'espace. Unigesture est un clavier gestuel car il a été conçu pour une interaction par inclinaison du périphérique (« tilting keyboard ») sans aucun retour visuel. La saisie est ambiguë, la désambiguïsation est faite par prédiction lexicale déclenchée après la saisie de l'espace (validation). L'absence de retour visuel impose de mémoriser l'agencement des caractères mais cela permet une adaptation sur des périphériques munis de petits écrans. TiltType (Partridge *et al.*, 2002) est une extension d'Unigesture et ajoute la possibilité de combiner des appuis boutons et les inclinaisons afin d'entrer du texte non ambigu. L'appui sur un ou plusieurs boutons entraîne l'affichage des caractères qui peuvent être sélectionnés en inclinant le périphérique. Cela facilite d'autant l'apprentissage. Les tests ont été réalisés sur une montre ce qui indique clairement que ce dispositif a principalement été introduit pour les périphériques de taille réduite en situation nomade.

(Költringer, *et al.*, 2007) ont proposé TwoStick utilisable dans un premier temps avec les deux joysticks d'un contrôleur de jeu. Tout comme T-Cube, il est basé sur une interaction à deux niveaux. L'espace des caractères est subdivisé en 9 zones de niveau 1, chaque zone étant subdivisée en 9 zones de niveau 2 (figure 58). Un caractère est associé à une zone de niveau 2 ce qui permet de représenter au maximum un alphabet de 81 caractères. Les 9 zones correspondent aux 8 directions et à la zone centrale d'un joystick. Ainsi, les 9 zones de niveau 1 correspondent à un joystick et les 9 zones de niveau 2 correspondent à l'autre joystick du contrôleur de jeu.

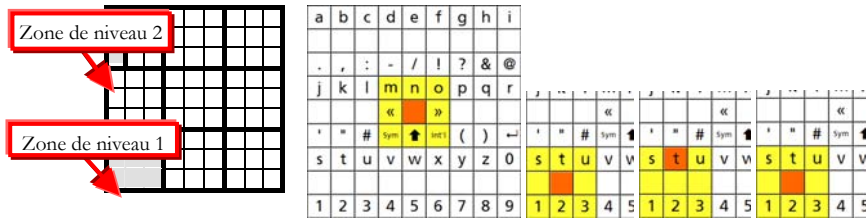


Figure 58 . Exemple d'interaction pour entrer le caractère 't' avec TwoStick

La figure 58 donne un exemple d'interaction. Au début, les deux joysticks sont centrés (couleur jaune pour la zone de niveau 1 et couleur orange pour la zone de niveau 2). La zone 7 de niveau 1 est alors sélectionnée avec le joystick de niveau 1. Ensuite, la zone 2 de niveau 2 (caractère 't') est sélectionnée avec le joystick de niveau 2. La validation du caractère est alors effectuée lorsque le joystick de niveau 2 revient au centre.

L'agencement des caractères doit être connu afin de pouvoir utiliser TwoStick. Un retour visuel est donc nécessaire au moins en phase d'apprentissage et l'agencement des caractères a été choisi afin de faciliter cet apprentissage. TwoStick présente toutefois l'avantage de supporter une interaction « tête levée » une fois cet agencement connu. La technique de validation impose une limitation à 72 caractères ce qui apparaît faible.

Alors que les solutions précédentes sont basées sur des interactions en deux dimensions, (KeyWheel, 2006) propose KeyWheel, un clavier basé sur une interaction à seulement une dimension. Son principe d'interaction est basé sur le suivi des séquences d'activation de zones ou touches contiguës. A l'origine, il a été conçu pour une utilisation sur des surfaces sensibles circulaires type iPod (iPod, 2007). Une séquence définir par un geste circulaire est ainsi obtenue à partir des touches sur lesquelles le doigt de l'utilisateur est passé. KeyWheel associe alors des opérations d'entrée à ces séquences.

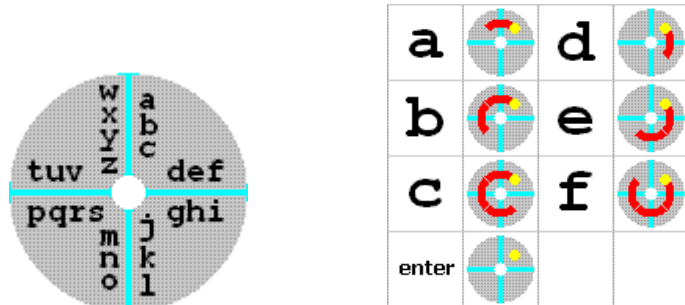


Figure 59. KeyWheel : les quatre zones d'interaction et exemple des séquences associées à la zone « abcdef »

La figure 59 présente un exemple de clavier alphabétique basé sur quatre touches. Chaque touche contient deux groupes de lettres. Afin de sélectionner une lettre, l'utilisateur doit commencer son geste sur la touche contenant la lettre et continuer par un geste circulaire autour du centre. La direction et la longueur de la rotation dépend de la lettre désirée. La figure 59 montre les gestes possibles pour obtenir les lettres de la touche « abcdef ». Cet agencement n'est qu'un exemple, il est possible d'utiliser plus ou moins de touches en fonction de la taille de la surface d'interaction. De même il est possible d'exploiter des gestes sur plus d'un tour

augmentant ainsi le nombre de lettres possibles pour une même touche de départ. Aucune expérimentation de ce dispositif n'est disponible à l'heure actuelle.

La saisie de texte chinois est très différente d'une saisie par lettres comme en Français ou en Anglais. (Ting-Chun Lai et Anderson, 2005) propose d'utiliser Donuts, une technique d'interaction gestuelle à base de menus en anneaux hiérarchiques. La particularité de l'approche est l'accès à certains menus en fonction de la pression du stylet. Un caractère chinois est défini par une syllabe dont les caractéristiques à définir sont : une consonne, une sous consonne (ou médian), une voyelle et un ton. Le clavier logiciel propose de sélectionner avec un stylet une consonne (figure 60-A). Les autres paramètres sont définis via les menus hiérarchiques. Le premier niveau de menus donne accès aux trois sous consonnes possibles (figure 60-B). Le second est divisé en deux menus (60-C et 60-E) pour conserver les performances des menus en anneaux : il ne faut pas plus de 8 éléments par menus et il y a 12 voyelles possibles en Chinois. Pour accéder au deuxième menu (60-E), il suffit d'exercer une première pression pour que le premier menu de voyelles se réduise (figure 60-D), puis exercer une pression plus forte (figure 60-E). Le troisième niveau de menu permet de sélectionner le ton.

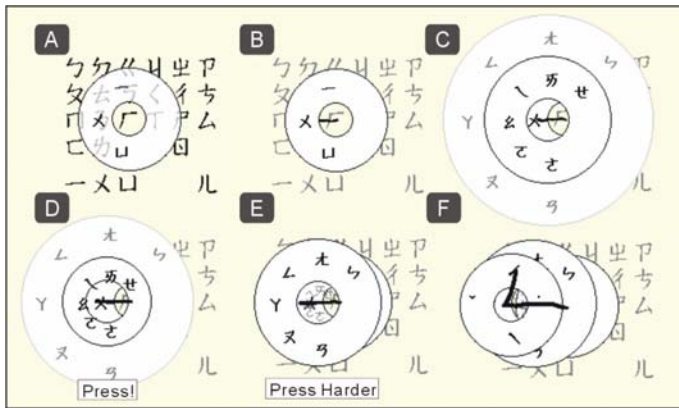


Figure 60. Menus en anneaux Donuts pour le Chinois

Le geste et les sélections effectuées sont visibles au centre de l'anneau afin d'aider l'utilisateur à comprendre sa sélection ainsi que le seuil de validation de sa sélection. Aucun test formel n'a été effectué (Ting-Chun Lai et Anderson, 2005).

Claviers gestuels vers une cible niveau mot

Contrairement aux claviers gestuels vers une cible niveau caractère, les claviers gestuels vers une cible niveau mot, appelés aussi « Word-level Unistroke », permettent à l'utilisateur d'écrire un mot sans lever son stylet (ou lâcher le bouton de la souris). L'analogie avec Unistroke est bien réelle : on obtient un mot par un et un seul « tracé » élémentaire, obtenu en reliant l'ensemble des lettres qui le composent. Tous les caractères disponibles sont accessibles directement. Dans l'hypothèse d'un affichage partiel, on circule entre plusieurs pages de même priorité, contenant les caractères, par l'intermédiaire d'un caractère-lien spécifique.

Le clavier Cirrin (Mankoff et Abowd, 1998) dispose dans un anneau l'ensemble des lettres de l'alphabet, de manière à minimiser la distance moyenne entre deux lettres. La saisie se fait mot à mot, en amenant successivement le pointeur sur chaque lettre du mot. Destiné à être employé avec un stylet, le système de saisie suppose que la main disponible (qui ne tient pas le stylet) pourra activer des touches

de contrôle pour la gestion des majuscules et de la ponctuation. La disposition des lettres dans l'anneau a été optimisée pour une saisie rapide en langue anglaise. Deux lettres seront très proches l'une de l'autre dans l'anneau si leur fréquence d'apparition côte à côte dans un mot est très élevée (figure 61).

Cette disposition n'est pas la plus facile à manipuler pour un débutant, qui risque d'être perdu et d'avoir du mal à trouver rapidement le caractère qu'il cherche. Elle a été définie pour permettre des performances plus élevées pour un utilisateur expérimenté (Mankoff et Abowd, 1998). Les distances entre deux lettres qui se suivent ont été raccourcies pour minimiser l'effet de la loi de Fitts. Pour tracer un mot, il suffit pour chaque lettre du mot (dans l'ordre où il s'écrit) d'entrer et sortir successivement de la case contenant la lettre. Il est possible de tracer des « boucles » pour sélectionner plusieurs lettres en même temps, comme la figure 61 le montre. Ces boucles sont rendues plus fréquentes par la disposition des lettres. Le mot est bien tracé en un seul geste continu. Lorsque le stylet est levé, le système considère que le mot est terminé et marque un espace. Cirrin ne fixe pas la taille de l'anneau et il ne peut pas être utilisé sans contrôle visuel.

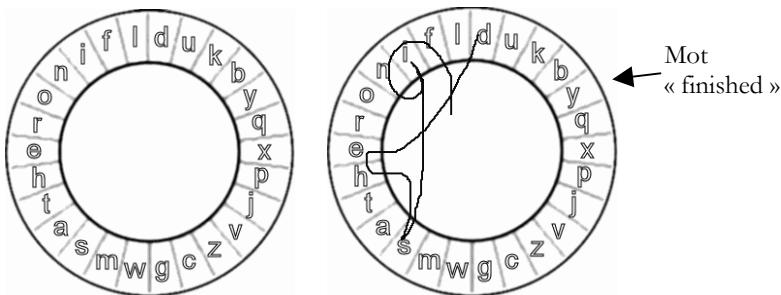


Figure 61. *Alphabet Cirrin*

Cirrin permet d'atteindre un niveau de rapidité de saisie correct, tout en offrant un retour visuel direct et complet. L'apprentissage sera certainement plus rapide qu'avec T-Cube, l'utilisateur ayant d'emblée l'ensemble des lettres déjà sous les yeux (le retour visuel est donc direct). Mais les performances risquent de plafonner car elles sont limitées par les trajets du stylet dans l'anneau ; ceci est frappant si on compare avec les gestes de T-Cube.

Le clavier ShapeWriter, anciennement connu sous le nom SHARK Shorthand-Aided Rapid Keyboarding (Zhai et Kristensson, 2003), se place au carrefour de deux technologies arrivant à maturité, les claviers à touches et les claviers gestuels. D'une part, les assistants numériques profitent tous d'une zone tactile, leur autorisant l'usage efficace d'un clavier gestuel. D'autre part les dispositions de touches des claviers à touches logiciels sont de plus en plus optimisées. Mais les claviers gestuels, s'ils semblent plus naturels, manquent de rapidité de même que l'écriture manuelle qu'ils imitent (Soukoreff et MacKenzie, 1995) et la saisie au stylet sur un clavier logiciel devient rapidement fastidieuse. D'où l'idée d'associer les deux types de claviers pour en tirer le meilleur. Ainsi, ShapeWriter est basé sur le clavier logiciel Atomik dont il utilise la disposition (Zhai *et al.*, 2002). Le geste tracé doit relier les touches du clavier Atomik qui composent le mot (figure 62). Pour associer un geste à un mot, le système implémente un algorithme de reconnaissance « élastique » détaillé dans (Kristensson et Zhai, 2004) et (Kristensson et Zhai, 2007), qui modélise le tracé pour chercher le symbole (et donc le mot) équivalent à un symbole « parfait ». Ce dernier sert de référence et est obtenu en passant par le

milieu de chaque touche composant le mot. Il s'agit donc bien d'un geste vers une cible.

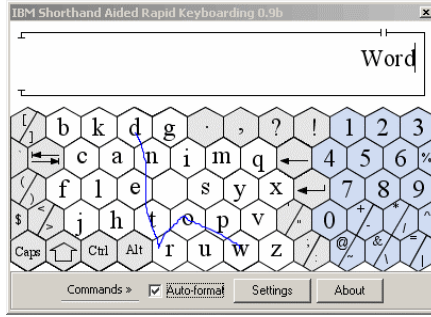


Figure 62. Première implémentation de ShapeWriter - SHARK : le symbole pour « word »

La première version de ShapeWriter, SHARK, présentée en figure 62 autorise deux types d'entrée. Si l'utilisateur ne maîtrise pas encore le clavier, il lui est possible de saisir le mot touche par touche comme avec un clavier logiciel standard. Puis, au fur et à mesure qu'il mémorise l'emplacement des touches, il glisse de plus en plus rapidement d'une lettre à l'autre, jusqu'à finir par tracer un geste complet reliant l'ensemble des lettres composant le mot. ShapeWriter met en œuvre ce principe d'apprentissage pour proposer un système proche de la sténographie ou des écritures à base de pictogrammes. À terme, l'utilisateur ne tracera plus que des symboles correspondant à des mots. Le but recherché est la désactivation du retour visuel qui libèrera l'espace et qui permettra un système de saisie similaire aux claviers à gestes symboliques vus précédemment. Mais on a ici l'avantage supplémentaire de la rapidité de saisie, car un geste code beaucoup plus de données. De plus, un tel système permettrait une saisie à l'aveugle que le clavier Atomik ne pouvait pas offrir.

On peut malgré tout remarquer que le symbole correspondant à un mot est défini par la disposition des touches du clavier. Ici cette disposition est de type Atomik, qui est optimisée pour la langue anglaise. Cette disposition n'est pas adaptée à la langue française, et par conséquent les gestes correspondant à des mots en français risquent d'être beaucoup plus compliqués. Mais la disposition des touches pourrait être d'un autre type, telle la disposition Qwerty proposée en supplément par le logiciel. Les symboles diffèrent alors par rapport à leur équivalent basé sur la disposition Atomik, et on perd l'optimisation apportée par cette dernière.

Aussi la seconde version de ce logiciel, appelée SHARK² (Kristensson et Zhai, 2004), règle-t-elle ce problème de disposition en masquant complètement la disposition des touches. SHARK² propose un geste pour chacun des mots courants de la langue anglaise et devient alors un des rares claviers gestuels symboliques de niveau « mot ». La figure 63 présente une fraction du dictionnaire. On pourra remarquer le nombre conséquent de symboles : un par mot.

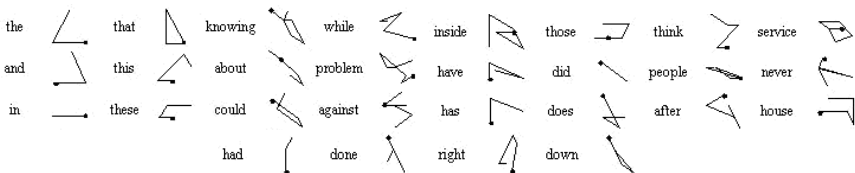


Figure 63. Quelques Symboles SHARK²

ShapeWriter est encore en phase d'étude. En proposant un symbole par mot, ce clavier ouvre une nouvelle voie de réflexion sur les méthodes de saisie informatiques. Ses concepteurs ont réussi leur pari de marier claviers logiciels avec claviers à gestes symboliques. SHARK² risque cependant de poser des problèmes aux utilisateurs, car le retour visuel est inexistant et l'utilisateur aura à apprendre le symbole correspondant à chaque mot. Ceci peut se faire par l'intermédiaire d'un dictionnaire papier, ou bien par apprentissage (Kristensson et Zhai, 2007). On se pose alors l'intérêt d'introduire un nouveau dictionnaire, sachant que la sténographie dispose d'un ensemble de symboles dont l'efficacité est prouvée de longue date.

Claviers gestuels vers une cible niveau phrase

Les claviers gestuels vers une cible niveau phrase permettent à l'utilisateur d'écrire un mot sans lever son stylet (ou lâcher le bouton de la souris). L'espace et les caractères de ponctuation sont accessibles au même titre de tous les autres caractères, il est donc inutile de valider chaque mot. Un texte complet peut être « tracé » en une seule fois.

Hex s'apparente à Unigesture en se basant sur des gestes continus (Williamson et Murray-Smith, 2005). Son alphabet est plus restreint : chaque symbole est représenté à l'aide de deux gestes élémentaires, un geste élémentaire étant un mouvement dans six directions possibles. Cela autorise donc 36 symboles différents. Une telle division conduit vers un espace pavé par des hexagones dans lequel l'utilisateur se déplace. L'interaction se décompose en deux étapes : la sélection d'un groupe de lettres et la sélection de la lettre proprement dite (figure 64). La saisie est non ambiguë. Hex repose sur une déformation non linéaire du pavage afin de favoriser certaines directions. Cette opération est réalisée grâce à un algorithme de prédiction de mots afin de faciliter certaines navigations et donc l'écriture de certaines lettres. Ce système est simple à utiliser et peut s'interfacier facilement avec de nombreux périphériques. En particulier, deux implémentations ont été réalisées à l'aide d'accéléromètres permettant de naviguer et donc d'écrire en inclinant le périphérique. Ce type d'interaction ressort comme une voie en plein développement pour les périphériques mobiles.

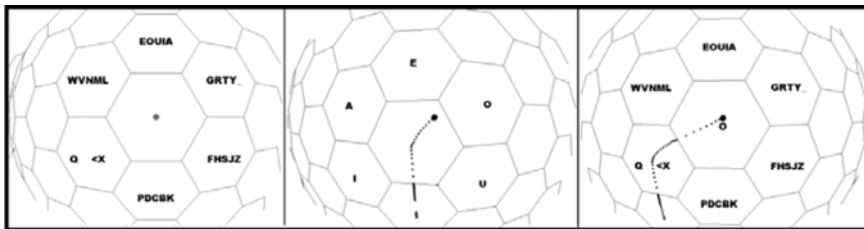


Figure 64. Hex : chaque lettre est assignée à un groupe qui est associé à un côté. Par exemple, la saisie d'un 'o' requiert un mouvement vers le haut suivi d'un mouvement vers le haut-droit.

QuikWriting (Perlin, 1998a) est un autre exemple de clavier gestuel vers une cible niveau phrase. Il a été conçu pour autoriser la saisie, sans lever le stylet, sans interrompre le mouvement. Bien entendu, l'utilisateur peut interrompre cette saisie s'il le désire. Afin de ne pas encombrer l'écran, l'ensemble des caractères est réparti sur plusieurs pages entre lesquelles on peut naviguer par un caractère spécial faisant office de lien. Chaque page est représentée par le système comme une grille de trois lignes par trois colonnes, où la zone centrale fait office de zone de repos et les autres zones contiennent les caractères (figure 65).

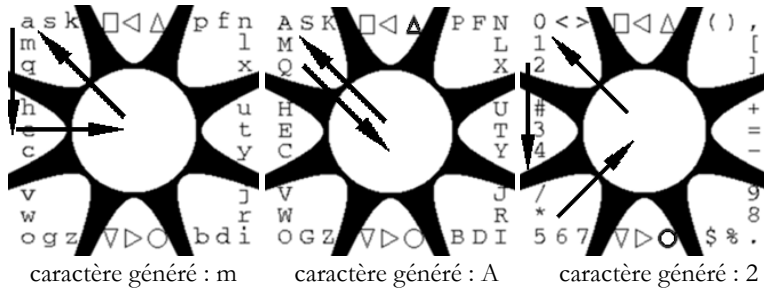


Figure 65. *Alphabet QuikWriting*

Pour obtenir un caractère, l'utilisateur fait un mouvement depuis la zone de repos vers la zone contenant le caractère qu'il désire, puis retourne vers la zone de repos en passant éventuellement par une des zones voisines. Les caractères sont répartis sur les huit zones périphériques de manière à ce que les plus fréquents soient accessibles le plus rapidement. Ainsi, pour les caractères Espace (le triangle qui pointe vers la gauche), 'o', 'e', 'a', 'n', 't' et 'i', qui sont les plus fréquents de la langue anglaise, il suffit de rentrer dans la zone correspondante puis d'en ressortir vers la zone de repos. Pour obtenir les autres caractères, il faut revenir dans la zone de repos en passant par les zones adjacentes les plus proches du caractère. Le nombre de zones à franchir correspond à la position du caractère à atteindre par rapport au caractère principal appartenant à la même zone. La figure 65 montre l'obtention du caractère 2 : il faut entrer dans sa zone, puis ressortir par ses deux voisins consécutifs les plus proches, car le caractère 2 est en deuxième position dans sa zone par rapport au caractère principal 0.

Initialement, QuikWriting présente la disposition à gauche de la figure 65, correspondant aux minuscules. On peut passer aux lettres majuscules en sélectionnant le triangle vers le haut, à la ponctuation en sélectionnant le carré, et aux nombres en sélectionnant le rond. Comme l'espace est un caractère au même titre que les autres, il devient possible d'entrer un texte complet sans interruption. Le concepteur de QuikWriting prévoit que, à terme, l'utilisateur puisse se passer de l'affichage de la disposition s'apparentant alors plus à un clavier symbolique.

On peut facilement envisager d'utiliser d'autres interfaces, aussi bien logicielles que matérielles (Isokoski et Raisamo, 2004). Diverses variantes ont été déclinées par (Perlin, 1998b), toujours basées sur une saisie au stylet. Le développement des outils initiés par QuikWriting se poursuit en partenariat avec Microsoft avec Xnav (Kanellos, 2006). L'agencement « optimisé » des caractères est à présent remplacé par un agencement alphabétique plus à même de faciliter l'apprentissage.

Dasher (Ward *et al.*, 2000) est un clavier qui associe cible niveau phrase et prédiction. Le choix du caractère se fait par un déplacement sur deux axes à l'aide d'un dispositif de pointage quelconque. En fonction de ce qui a déjà été saisi, l'agencement des caractères varie pour mettre en évidence les caractères qui ont le plus de chance d'être sélectionnés. L'interface est étudiée pour guider l'utilisateur vers les caractères les plus probables grâce à la prédiction. On peut le voir sur la figure 66, les caractères les plus probables sont affichés dans un bloc plus gros. L'efficacité de ce clavier dépend du degré de finesse de l'algorithme de prédiction qui semble excellent.

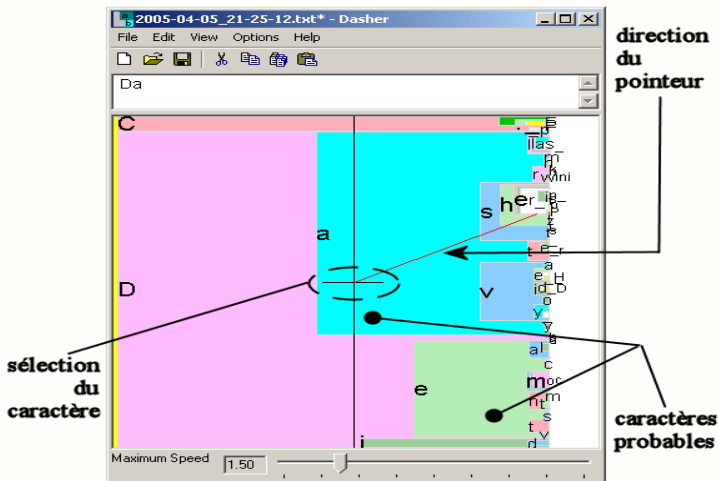


Figure 66. Dasher en pleine exécution

Les caractères défilent, et on se déplace verticalement pour aller à la rencontre du caractère désiré. Le trait sur la barre verticale au milieu symbolise le pointeur qui « attrape » les caractères, et le texte saisi s'affiche dans le champ de texte en haut de la fenêtre. On peut contrôler la vitesse du défilement en déplaçant horizontalement le dispositif de pointage. Ceci est un plus pour les utilisateurs novices, qui iront moins vite pour mieux contrôler leur saisie. Dasher offre une aide au pointage, matérialisée par une ligne rouge qui suit les mouvements du dispositif de pointage. Cette aide vise à rendre la sélection des caractères plus aisée et moins fatigante pour les yeux. Initialement, toutes les lettres ont la même priorité et sont toutes dans le « champ de vision ». Mais au fur et à mesure qu'on se rapproche d'une lettre, les autres deviennent invisibles et il est nécessaire de revenir en arrière pour en choisir une autre. Ce clavier propose un retour visuel incomplet mais direct, et peut s'adapter à toutes sortes de périphériques de pointage tout en permettant des performances plus que correctes. Cependant, la saisie peut apparaître fatigante à l'usage : on est obligé de fixer constamment le pointeur pour le diriger parmi l'ensemble des lettres et les lettres sont présentées sur des blocs dont les couleurs s'alternent rapidement pour mettre en évidence les choix, ce qui fatigue rapidement les yeux.

3.4 Performances des claviers gestuels

Le tableau 2 donne les performances des différents claviers gestuels présentés précédemment.

Les performances sont données à titre indicatif en se basant sur différentes études. Il est toutefois difficile de trouver des tests exhaustifs concernant la totalité ou même une partie des claviers examinés dans cette section. D'une part, la durée du protocole de tests ainsi que le protocole de tests lui-même sont très difficilement comparables entre les différents tests. D'autre part, les tests réalisés ne prennent pas forcément en compte les caractères accentués ou la ponctuation. Le tableau 2 reprend la même définition pour le niveau des utilisateurs que le tableau 1. Enfin, à chaque fois que cela est possible ou nécessaire, la technique d'interaction ou le périphérique utilisé pour l'interaction est précisé.

Type	Claviers	Vitesse (wpm)		
		Novice	Expert	
Claviers gestuels symboliques	Unistroke		34 théorique (Isokoski, 1999)	
	Graffiti	7,5 (Fleetwood <i>et al.</i> , 2002)	21,5 après 3 mois d'utilisation (Fleetwood <i>et al.</i> , 2002)	
	Graffiti 2	6,64 (commandes) 7,52 (chiffres) 9,24 (lettres) (Költringer et Grechenig, 2004)	inconnu	
	MDITIM	2,4 (Isokoski, 1999), touchpad	7,6 après 5 heures (Isokoski, 1999), touchpad	
	EdgeWrite	6,6 (Wobbrock, 2006), stylet 6,4 (Wobbrock, 2006), gamepad	24 par un des auteurs (Wobbrock, 2006), stylet 14,7 par un des auteurs (Wobbrock, 2006), gamepad	
	Thumbscript	inconnu	10-20 théorique (Thumbscript Development, 2000), touches	
	Symbol Creator	inconnu	8,58 après 2h30 (Miniotas <i>et al.</i> , 2003), regard	
	Glyph	2,5 (Poirier et Belatar, 2006b)	4 après quelques minutes (Poirier et Belatar, 2006b), touches	
	Glyph 2	4 (Poirier et Belatar, 2006b)	7 après quelques minutes (Poirier et Belatar, 2006b), touches	
	UniGlyph	~5,5 (Poirier et Belatar, 2007), stylet sur PDA	12,5 après ~15 minutes (Poirier et Belatar, 2007), stylet sur PDA	
Claviers gestuels vers une cible	niveau caractère	T-Cube	16 (Venolia et Neiberg, 1994)	
		VirHKey	6,6 (Martin, 2005b)	
		Unigesture	4-6 (Sazawal <i>et al.</i> , 2002)	
		TiltType	inconnu	
		TwoStick	5,1 (Költringer <i>et al.</i> , 2007)	
		KeyWheel	inconnu	
	niveau mot	Cirrin	inconnu	20 après 2 mois d'utilisation (Mankoff et Abowd, 1998), stylet
		SHARK ²	7 (Cockburn <i>et al.</i> , 2007), souris	51,6-85,6 par les auteurs pour des mots connus (Kristensson et Zhai, 2004), souris
		Hex	inconnu	10-12 après 30h (Williamson et Murray-Smith, 2005), PDA inclinable
	niveau phrase	QuickWriting	4 (Isokoski et Raisamo, 2004), stylet et gamepad	16 après 10h (Isokoski et Raisamo, 2004), stylet 13 après 10h (Isokoski et Raisamo, 2004), gamepad
		Dasher	4-14 (Ward <i>et al.</i> , 2000), souris	12-26 après 6h (Ward <i>et al.</i> , 2000), souris 34 par un des auteurs (Ward <i>et al.</i> , 2000), souris

Tableau 2. Performance des claviers gestuels

De manière générale, les claviers gestuels sont limités par la vitesse d'écriture manuscrite que l'on situe généralement autour de 30 mots par minute. Les protocoles de test influent également énormément les résultats présentés. La prise en compte ou non de l'erreur de saisie peut à elle seule expliquer les différences entre les claviers. Il est intéressant de noter les résultats très proches de la plupart

des claviers à gestes. Un premier groupe se forme autour de 10 wpm ce qui est assez faible et un second groupe ressort autour de 20 wpm. Seul Shark² et Dasher se détachent particulièrement. Leurs gestes n'étant pas associés nécessairement à un symbole par caractère, cela permet d'augmenter sensiblement les performances. Cependant, en ce qui concerne Shark², les tests menés ont été réalisés sur un nombre très restreint de mots et ne reflètent pas une utilisation réelle. Il est à noter que plusieurs claviers à gestes comme par exemple EdgeWrite, Glyph ou VirHKey peuvent apparaître dans d'autres catégories de claviers car différentes techniques d'interaction peuvent être utilisées.

4 Conclusion

Il est probable que le clavier standard Azerty/Querty a encore de beaux jours devant lui, car encore maintenant tous les ordinateurs en France sont vendus avec un clavier Azerty. Le changement viendra peut-être des utilisateurs, qui finiront par adopter des systèmes informatiques prenant de moins en moins de place. Ces systèmes seront certainement munis d'un écran tactile et dépourvus de clavier physique.

Les claviers logiciels standard proposent un retour visuel complet et sont familiers de beaucoup d'utilisateurs. Cependant ils prennent trop de place pour être affichés intégralement sur un PDA et on a aussi montré qu'ils constituent un anachronisme, car la disposition qu'ils utilisent n'a plus lieu d'être ; sachant qu'il existe d'autres dispositions bien plus à même d'être efficaces.

Les claviers logiciels optimisés soit pour la performance soit pour une meilleure facilité d'emploi sont efficaces dans le but recherché, mais tout aussi encombrants à l'écran que les précédents. De plus il faut relativiser ces résultats car souvent ils reposent sur des évaluations théoriques optimistes, soient sur des évaluations de claviers incomplets, soient sur la saisie de mots ou de phrases types. Enfin, l'apprentissage nécessaire d'un nouvel agencement rebute souvent l'utilisateur novice. D'où l'idée de se défaire complètement du modèle de clavier physique par l'introduction de claviers à gestes plus proches de l'écriture manuscrite.

Les claviers à gestes symboliques sont populaires auprès des utilisateurs car ils se rapprochent de l'écriture manuelle. Or, ils sont plus lents et n'offrent pas de retour visuel, si ce n'est le caractère qui s'ajoute au texte. De plus, ils nécessitent un apprentissage parfois lourd pour arriver à de bonnes performances. En contrepartie, pour la plupart d'entre eux la saisie à l'aveugle est possible, ce qui est très agréable pour la prise de notes. D'un autre côté, les claviers à sélection offrent des performances supérieures aux claviers à gestes symboliques, et fournissent un retour visuel à l'utilisateur. Mais ils ne sont pas toujours intuitifs et la saisie doit pour la plupart s'effectuer à un emplacement bien précis, forçant l'utilisateur à contrôler visuellement chaque geste.

Il est clair que si on va vers un affranchissement de l'encombrant clavier physique, il n'est pas dit que ce dernier n'ait qu'un seul remplaçant universel, et (Isokoski, 1999) a d'ailleurs bien su le montrer. Il est probable, au vu de l'intéressant marché que cela ouvre, que l'on va vers une multiplication croissante des systèmes de saisie, et ce au bénéfice de l'utilisateur. Tous les claviers existants n'ont d'ailleurs pas pu être cités dans ce document, tant leur nombre est grand.

Certaines technologies semblent vouées à un avenir certain. Le concept multi-tactile où l'utilisateur peut utiliser non plus un seul doigt en simultané mais plusieurs doigts offre de nouvelles possibilités même si cette technologie est assez ancienne (Buxton, 2007). La mise sur le marché du premier périphérique « grand public »

utilisant le multi-tactile devrait ouvrir la voie à d'autres résultats (Apple, 2008). Les systèmes de prédiction apparaissent également comme un composant incontournable des techniques « modernes » d'entrée de texte. Il permet une réduction de la taille du clavier en surface ou en nombre de touches et surtout, il autorise une grande robustesse à l'erreur. Cependant de nombreuses recherches restent nécessaires afin de rendre son intégration plus conviviale dans le système de saisie.

Remerciements

Nous remercions vivement Xavier Richez qui a fourni un important travail sur cet état de l'art lors de son travail de recherche.

5 Bibliographie

Apple, (2008). *iPhone*. Disponible à : <http://www.apple.com/fr/iphone/>

Ballayer, R. (2005). *Guide d'accès à l'informatique auprès du public handicapé moteur en Pays de Loire*. Disponible à : www.handitec.com/infoservices/Doc/guide_AT_informatique16mars2005.pdf

Belatar, M. (2005). Entrée de données pour les systèmes interactifs nomades. GLYPH : écrire avec 7 touches en un seul geste. Master report. Disponible à : www-valoria.univ-ubs.fr/Mohammed.Belatar/docs/RAPPORT_DEA.pdf

Bellaire Electronics (2006). *The Cykey*. Disponible à : www.bellaire.demon.co.uk/bellaire_cykey_about.html

Bellman, T., MacKenzie, I.S. (1998). A probabilistic character layout strategy for mobile text entry. In *proceedings of Graphics Interface '98*, 168-176.

Boissière, P., Schadle, I. (2006). Proposition d'un cadre méthodologique d'évaluation des systèmes d'assistance à la saisie de textes : Applications aux systèmes Sibylle et VITIPI. In *Handicap 2006*, publié dans AMSE, Paris, 07-09 Juin, 149-154.

Brook, M. (2000). *Introducing the Dvorak Keyboard*. Disponible à : <http://www.mwbrooks.com/dvorak/index.html>

Bunke, H. (2003). Recognition of Cursive Roman Handwriting - Past, Present and Future. In *Proceedings of the Seventh international Conference on Document Analysis and Recognition (ICDAR)*, Vol. 1, August 03 - 06, IEEE Computer Society, Washington, DC, 448-459.

Buxton, B. (2007). *Multi-Touch Systems that I Have Known and Loved*. Disponible à : <http://www.billbuxton.com/multitouchOverview.html>

Carey, D. (2004). *FrogPad iFrog Bluetooth Keyboard*. Gadgeteer Hands On Review. Disponible à : www.thegadgeteer.com/review/frogpad_ifrog_bluetooth_keyboard_review

Case, L. (2004). Ergodex DX1 Redefines Input Devices. *Game Developers Conference 2004*, ExtremeTech special report. Disponible à : <http://www.extremetech.com/article2/0,1697,1555134,00.asp>

Castellucci, S.J. (2007). *Mobile Text Entry Using Ambiguous Keypads: New Metrics In A New Toolkit*. M.Sc. Thesis, York University, May.

- Chau, D.H., Wobbrock, J.O., Myers, B.A., Rothrock, B. (2006). Integrating isometric joysticks into mobile phones for text entry. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, Montréal, Québec, Canada, April 22-27, ACM Press, New York, NY, 640-645.
- Clarkson, E., Clawson, J., Lyons, K., Starner, T. (2005). An empirical study of typing rates on mini-QWERTY keyboards. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, Portland, OR, USA, April 02-07, ACM Press, New York, NY, 1288-1291.
- Claviature (2006). <http://www.microth.com/claviature/>
- Clawson, J., Lyons, K., Starner, T., Clarkson, E. (2005). The Impacts of Limited Visual Feedback on Mobile Text Entry for the Twiddler and Mini-QWERTY Keyboards. In *Proceedings of the Ninth IEEE International Symposium on Wearable Computers* (October 18-21, 2005), ISWC, IEEE Computer Society, Washington, DC, 170-177.
- Cockburn, A., Siresena, A. (2003). Evaluating Mobile Text Entry with the Fastap Keypad. In *People and Computers XVII (Volume 2): British Computer Society Conference on Human Computer Interaction*, Bath, England, 77-80.
- Cockburn, A., Kristensson, P.O., Alexander, J., Zhai, S. (2007). Hard lessons: effort-inducing interfaces benefit spatial learning. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '07)*, ACM Press, 1571-1580.
- CooTek. (2008). *TouchPal*. Disponible à : <http://www.cootek.com/>
- Eatoni (2007). <http://www.eatoni.com>
- Endfield, C. (1978). A typewriter in your pocket. *Computer Weekly*, 11 May, 4.
- Engelbart, D.C. (1968). A Research Center for Augmenting Human Intellect (90-min. video recording), live online hypermedia demonstration/presentation at the *Fall Joint Computer Conference*, San Francisco, CA, December 9, 1968.
- Evreinova, T., Evreinov, G., Raisamo, R. (2004). Four-Key Text Entry for Physically Challenged People. In *Proceedings of the 8th ERCIM Workshop "User Interfaces For All"*, 28-29 June, Vienna, Austria, 12 pages. Disponible à : www.ui4all.gr/workshop2004/files/ui4all_proceedings/adjunct/techniques_devices_metaphors/16.pdf
- Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47, 381-391.
- Fleetwood, M.D., Byrne, M.D., Centgraf, P., Dudziak, K., Lin, B., Mogilev, D. (2002). An analysis of text-entry in Palm OS—Graffiti and the Virtual Keyboard. In *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting*, 617-621.
- FrogPad (2006), <http://www.frogpad.com/>.
- Goldberg, D., Richardson, C. (1993). Touch-typing with a stylus. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI'93)*, ACM Press, New York, NY, USA, 80-87.
- Gong, J., Haggerty, B., Tarasewich, P. (2005). An enhanced multitap text entry method with predictive next-letter highlighting. In *CHI '05 Extended Abstracts on*

Human Factors in Computing Systems, Portland, OR, USA, April 02-07, ACM Press, New York, NY, 1399-1402.

Gong, J., Tarasewich, P. (2005). Testing Predictive Text Entry Methods with Constrained Keypad Designs. In *Proceedings of the 11th International Conference on Human-Computer Interaction (HCI05)*, Las Vegas, Nevada USA. Disponible à : www.ccs.neu.edu/home/tarase/HCI05_GongTarase.pdf

Harbusch K., Hasan, S., Hoffmann, H., Kühn, M., Schüler, B. (2003). Domain-specific Disambiguation for Typing with Ambiguous Keyboards. In *EACL 2003 Workshop on Language Modeling for Text Entry Methods*, Budapest, April 14. Disponible à : <http://www.uni-koblenz.de/~harbusch/harbusch-et-al-eacl-2003.pdf>.

Hick, W. E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, 4, 11-26.

Hopkins, D. (1991). The Design and Implementation of Pie Menus. *Dr. Dobb's Journal*, December, 16-26.

Hyman, R. (1953). Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*, 45, 423-432.

In10did (2005). <http://www.in10did.com/>

iPod (2007), <http://www.apple.com/fr/ipod/ipod.html>

Isokoski, P. (1999). *A Minimal Device-Independent Text Input Method*. Disponible à : www.cs.uta.fi/~poika/g/g.html.

Isokoski, P. (2001). Model for unistroke writing time. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI2001)*, ACM Press, New York, NY, USA, 357-364.

Isokoski, P. (2004). Performance of menu-augmented soft keyboards. In *Proceeding of ACM Conference on Human Factors in Computing Systems, CHI Letters*, 6(1), 423-430.

Isokoski, P., Raisamo, R. (2004). QuikWriting as a Multi-Device Text Entry Method. In *Proceedings of the third Nordic conference on Human-computer interaction*, ACM Press, New York, NY, USA, 105-108.

James, C. L., Reischel, K. M. (2001). Text input for mobile devices: comparing model prediction to actual performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'01)*, Seattle, Washington, United States, ACM Press, New York, NY, 365-371.

Jhaveri, N. (2003). Two Characters per Stroke – A Novel Pen-Based Text Input Technique. In G. Evreinov (ed.), *New Interaction Techniques'03*, Spring, University of Tampere Finland, 10-15. Disponible à : www.cs.uta.fi/reports/bsarja/B-2003-5.pdf

Kanellos, M. (2006). Microsoft scientists pushing keyboard into the past. *CNET News.com*, May 3.

KeyTouch Corp. (2003). *MultiPlay Keyboard Software*. Disponible à : <http://www.keytouch.net/products.html>.

KeyWheel (2006), <http://www.microth.com/circumscrip/>

Kober, H., Skepner, E., Jones, T., Gutowitz, H., MacKenzie, S. (2001). Linguistically optimized text entry on a mobile phone. Disponible à : www.eatoni.com/research/chi.pdf.

Költringer, T., Grechenig, T. (2004). Comparing the immediate usability of graffiti 2 and virtual keyboard. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, Vienna, Austria, April 24 - 29, ACM Press, New York, NY, 1175-1178.

Költringer, T., Isokoski, P., Grechenig, T. (2007). TwoStick: writing with a game controller. In *Proceedings of Graphics interface 2007 (GI'07)*, Montreal, Canada, May 28-30, vol. 234, ACM Press, New York, NY, USA, 103-110.

Krakow, G. (2000). *Faster with Fitaly*. Disponible à : www.fitaly.com/fitalycomment/fitalystampstart.htm

Kristensson, P.-O., Zhai, S. (2004). SHARK²: A Large Vocabulary Shorthand Writing System for Pen-based Computers. In *Proceeding of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST 2004)*, ACM Press, New York, NY, USA, 43-52.

Kristensson, P.-O., Zhai, S. (2007). Learning shape writing by game playing. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '07)*, ACM Press, 1971-1976.

Land, R.I. (1981). Keyboard entry - can it be simplified? In *Proceedings of the joint conference on Easier and more productive use of computer systems. (Part - II): Human interface and the user interface*, ACM Press, New York, NY, USA, 53-58.

Lebedev, A. (2005). *Optimus keyboard*. Disponible à : www.artlebedev.com/portfolio/optimus/

Lee, P., Zhai, S. (2004). Top-down learning strategies: can they facilitate stylus keyboard learning? In *International Journal of Human-Computer Studies*, Vol. 60 (5-6), 585-598.

Lyons, K., Plaisted, D., Starner, T. (2004a). Expert Chording Text Entry on the Twiddler One-Handed Keyboard. In *Proceedings of the Eighth international Symposium on Wearable Computers (Iswc'04)*, October 31 - November 03, Washington, DC. IEEE Computer Society, 94-101.

Lyons, K., Starner, T., Plaisted, D., Fusia, J., Lyons, A., Drew, A., et Looney, E. W. (2004b). Twiddler typing: one-handed chording text entry for mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*, Vienna, Austria, April 24 - 29, ACM Press, New York, NY, 671-678

Lyons, K., Gane, B., Starner T., Catrambone, R. (2005). Improving Novice Performance on the Twiddler One-Handed Chording Keyboard. In *Proceedings of the International Forum on Applied Wearable Computing*, Zurich, Switzerland, March. Disponible à : www.cc.gatech.edu/ccg/publications/twiddler-novice.pdf.

MacKenzie, I.S., Zhang, S.X., Soukoreff, R.W. (1999). Text entry using soft keyboards. *Behaviour & Information Technology*, 18, 235-244.

MacKenzie, I.S., Zhang, X.S. (1999). The design and evaluation of a high performance soft keyboard. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, ACM Press, New York, NY, USA, 25-31.

- MacKenzie, I.S., Kober, H., Smith, D., Jones, T., Skepner, E. (2001). LetterWise: Prefix-based disambiguation for mobile text input. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2001)*, ACM Press, New York, NY, USA, 111-120.
- MacKenzie, I.S. (2002a). *A Note on Calculating Text Entry Speed*. Disponible à : <http://www.yorku.ca/mack/RN-TextEntrySpeed.html>.
- MacKenzie, I.S. (2002b). KSPC (keystrokes per character) as a characteristic of text entry techniques. In *Proceedings of the Fourth International Symposium on Human-Computer Interaction with Mobile Devices*, 195-210, Heidelberg, Germany, Springer-Verlag.
- MacKenzie, I.S., Soukoreff, R.W. (2002). Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction*, 17, 147-198.
- Magnien, L., Bouraoui, J-L. Vella, F. (2003). Utilisation d'indices visuels pour l'aide à la saisie de texte sur PDA. *Actes de la Conférence Francophone sur l'Interaction Homme-Machine*, Caen, France, 25-28 Novembre, 252-255.
- Magnien, L., Bouraoui, J.L., Vigouroux, N. (2004). Mobile devices: soft keyboard text-entry enhanced by Visual Cues. In *Proceedings of the 1st French-Speaking Conference on Mobility and Ubiquity Computing (UbiMob '04)*, Nice, France, June 01-03, vol. 64. ACM Press, New York, NY, 158-165.
- Mankoff, J., Abowd, G.D. (1998). Cirrin: a word-level unistroke keyboard for pen input. In *Proceeding of the 11th Annual ACM Symposium on User Interface Software and Technology (UIST 1998)*, ACM Press, New York, NY, USA, 213-214.
- Martin, B. (2005a). VirHKey: un clavier gestuel hyperbolique avec retour visuel. In *Actes de la Conférence Francophone sur l'Interaction Homme-Machine*, Toulouse, France, 27-30 Septembre, 19-26.
- Martin, B. (2005b). VirHKey: a VIRtual Hyperbolic KEYboard with gesture interaction and visual feedback for mobile devices. In *Proceeding of MobileHCI 2005, 7th International Conference on Human Computer Interaction with Mobile Devices and Services*, 19 to 22 September 2005, University of Salzburg, Austria, 99-106.
- Martin, B., Isokoski, P. (2008). Edgewrite with Integrated Corner Sequence Help. In *SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, Florence, Italy, April 5-10, ACM Press, New York, NY, USA, to appear.
- Masui, T. (1999). POBox: An Efficient Text Input Method for Handheld and Ubiquitous Computers. In *Proceedings of the International Symposium on Handheld and Ubiquitous Computing (HUC'99)*, September, 289-300.
- Matias, E., MacKenzie, I. S., Buxton, W. (1996). One-Handed Touch-Typing on a QWERTY Keyboard. *Human-Computer Interaction*, Vol. 11, No. 1, 1-27.
- Miniotas, D., Spakov, O., Evreinov G. (2003). Symbol Creator: An Alternative Eye-based Text Entry Technique with Low Demand for Screen Space. In *Proceeding of Human-Computer Interaction - INTERACT'03*, M. Rauterberg et al. (Eds.), IOS Press, (c) IFIP, 137-143.
- Nesbat, S.B. (2003). A System for Fast, Full-Text Entry for Small Electronic Devices. In *Proceedings of the Fifth International Conference on Multimodal Interfaces, ICMI 2003* (ACM-sponsored), Vancouver, November 5-7, 2003. MessagEase. Disponible à : www.exideas.com/ME/index.html

Oniszcak, A., MacKenzie, I.S. (2004). A Comparison of Two Input Methods for Keypads on Mobile Devices. In *Proceeding of NordiCHI 2004*, October 23-27, Tampere, Finland, 101-104.

Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G., Want, R. (2002). TiltType: accelerometer-supported text entry for very small devices. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (UIST'02)*, Paris, France, October 27-30, ACM Press, New York, NY, 201-204.

Pavlovych, A., Stuerzlinger, W. (2004). Model for non-expert Text Entry Speed on 12-Button Phone Keypads. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI04)*, ACM Press, New York, NY, USA, 351-358.

PCD Maltron LTD (2004). Let's talk sense about keyboard induced pain. In *BCS (British Computer Society)*. Disponible à : archive.bcs.org/BCS/review04/articles/mobilecomputing/keyboardinducedpain.htm

Perlin, K. (1998a). Quikwriting: continuous stylus-based text entry. In *Proceeding of the 11th Annual ACM Symposium on User Interface Software and Technology (UIST 1998)*, ACM Press, New York, NY, USA, 215-216.

Perlin, K. (1998b). Quikwriting. Disponible à : <http://mrl.nyu.edu/~perlin/experiments/quikwriting/>

Plamondon, R. Srihari, S.N. (2000). On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22, 1 January, 63-84.

Poirier, F., Schadle, I. (2004). État de l'art des méthodes de saisie de données sur dispositifs nomades. Typologie des approches. In *Actes de la Conférence Francophone sur l'Interaction Homme-Machine*, ACM Press, New York, NY, USA, 133-140.

Poirier, F. (2005). Glyph: A New Stroke-Alphabet for Stylus-Based or Key-Based Text Entry. In *Proceeding of 11th Conference on Human-Computer Interaction - HCI 2005*, Las Vegas, USA. Disponible à : dmt.fh-joanneum.at/kd3/objects/application_pdf/2870_glyph.pdf

Poirier, F., Schadle, I. (2005). Interfaces réduites pour l'entrée de texte - État de l'art et présentation du communicateur Sibylle pour l'aide à l'écriture de personnes handicapées. *Revue d'Interaction Homme-Machine*, Vol. 6, No2, 63-89.

Poirier, F., Belatar, M. (2006a). Evaluation d'analogies scripturales pour la conception d'une méthode de saisie en mobilité - Uni-Glyph. In *Ergo'LA 2006*, ESTIA & ESTIA.INNOVATION (ed.), Biarritz, October, 333-336.

Poirier, F., Belatar, M. (2006b). Glyph 2 : une saisie de texte avec deux appuis de touches par caractère - Principes et comparaisons. In *18th int. conf. on Association Francophone d'Interaction Homme-Machine (IHM)*, Montréal, Canada, 18/04-21/04, Vol. 133, ACM Press, 159-162.

Poirier, F., Belatar, M. (2007). UniGlyph : une méthode universelle pour la saisie de texte sur dispositifs mobiles. In *19th int. conf. on Association Francophone d'Interaction Homme-Machine (IHM)*, Paris, France, 13-15/11, ACM Press, 111-118.

Polital Enterprises (2003). *CNTDesigner Keyboard Designer*. Disponible à : <http://www.polital.com/cntd/>

- Pratt, F. (1939). *Secret and Urgent: the Story of Codes and Ciphers*. Blue Ribbon Books.
- Raynal, M., Vigouroux, N. (2005a). KeyGlasses: Semi-transparent keys to optimize text input on virtual keyboard. Dans : *Assistive technology: from virtuality to reality - 8th European conference for the advancement of assistive technology in Europe (AAATE 2005)*, Lille, 06-09 Septembre, 713-717.
- Raynal, M., Vigouroux, N. (2005b). Genetic Algorithm to Generate Optimized Soft Keyboard. In *Int. Conf. for Human-Computer Interaction (CHI 2005)*, Portland, Oregon USA, 02/04/05-07/04/05, ACM Press, 1729-1732.
- Raynal, M. (2006a). Claviers GAG : claviers logiciels optimisés pour la saisie de texte au stylet. In *18th int. conf. on Association Francophone d'Interaction Homme-Machine (IHM)*, Montréal, Canada, 18/04-21/04, Volume 133, ACM Press, 3-10.
- Raynal, M. (2006b). Le système KeyGlass : aide à la saisie de caractères pour personnes handicapées moteur. Retours sur une première expérimentation. *Handicap 2006*, Paris, 07/06-09/06, 155-160.
- Raynal, M., Truillet, P. (2006). Le clavier FishEye. In *18th int. conf. on Association Francophone d'Interaction Homme-Machine (IHM)*, Montréal, Canada, 18/04-21/04, Volume 133, ACM Press, 289-290.
- Raynal, M., Truillet, P. (2007). Fisheye Keyboard: Whole Keyboard Displayed on PDA. *HCI International*, Beijing (China), 22/07-27/07, Vol. 2, J. Jacko (Eds.), Springer-Verlag, LNCS 4551, 452-459.
- Roeber, H., Bacus, J., Tomasi, C. (2003). Typing in Thin Air, The Canesta Projection Keyboard – A New Method of Interaction with Electronic Devices. In *Int. Conf. for Human-Computer Interaction (CHI 2003)*, Ft. Lauderdale, Florida, USA, April 5-10, ACM Press, 712-713.
- Ryu, H., Cruz, K. (2005). LetterEase: Improving text entry on a handheld device via letter reassignment. In *Proceedings of the 19th Conference of the Computer-Human interaction Special interest Group (Chisig) of Australia on Computer-Human interaction: Citizens online: Considerations For Today and the Future*, Canberra, Australia, November 21–25, ACM International Conference Proceeding Series, vol. 122, Computer-Human Interaction Special Interest Group (CHISIG) of Australia, Narrabundah, Australia, 1-10.
- Sandnes, F.E., Aubert, A. (2007). Bimanual Text Entry using Game Controllers: Relying on Users' Spatial Familiarity with QWERTY. *Interacting with Computers*, Vol. 19, No. 2, March, 140-150.
- Sazawal, V., Want, R., Borriello, G. (2002). The Unigesture Approach. In *Proceedings of the Mobile Human-Computer Interaction : 4th International Symposium (Mobile HCI 2002)*, Pisa, Italy, September 18-20, LNCS, Springer Verlag, 485-496.
- Smith, J.M.D (2000). *Thumbscript™ : Designing a general solution to the problem of text input in small devices*. Disponible à : <http://www.thumbscript.com/Downloads/TWhitePaper.pdf>
- Softava Company (2000). *SilkyBoard - Final release version*. Disponible à : softava.com/q12.
- Soukoreff, R.W., MacKenzie, I.S. (1995). Theoretical upper and lower bounds on typing speed using a stylus and soft keyboard. In *Behaviour & Information Technology*, 14, 370-379.

Soukoreff, R.W. (2002). *Text entry for mobile systems: Models, measures, and analyses for text entry research*. M.Sc. Thesis, York University, June.

SpeedScript (1999). <http://www.speedscript.biz/>

Thomasset, T. (1999). *Tout sur les unités de mesure*. Disponible à : <http://www.utc.fr/~tthomass/Themes/Unites/index.html>

Thumbscript Development, LLC. (2000). *Frequently Asked Questions*. Disponible à : <http://www.thumbscript.com/faqs.html>

Tiainen, S. (2000). *The global keyboard optimised for small wireless devices*. GKOS. Disponible à : gkos.com.

Ting-Chun Lai J., Anderson R., Li Y. (2005). Donuts: Chinese Input with Pressure-Sensitive Marking Menu. In *Conference Supplement to UIST'2005: Demonstration*, Seattle, WA. Disponible à : dub.washington.edu/pubs/uist2005/dounts.pdf.

Vella, F., Vigouroux, N. (2004). Etude de l'optimisation de claviers virtuels au travers de sujets handicapés. *Secondes Rencontres Jeunes Chercheurs en Interaction Homme-Machine (RJC-IHM 2004)*, Lacanau, 20-22 octobre, 99-102.

Vella, F., Vigouroux, N., Truillet, P. (2004). Environnement de conception de clavier virtuel : SOKEYTO (Software KEYboard TOolkit). In *Actes de la Conférence Francophone sur l'Interaction Homme-Machine*, ACM Press, New York, NY, USA, 181-182.

Venolia, D., Neiberg, F. (1994). T-Cube: a fast, self-disclosing pen-based alphabet. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI94)*, ACM Press, New York, NY, USA, 265-270.

Ward, J.D., Blackwell, A.F., MacKay, D.J.C. (2000). Dasher - a data entry interface using continuous gestures and language models. In *Proceedings of the 13th annual ACM symposium on User Interface Software and Technology (UIST 2000)*, ACM Press, New York, NY, USA, 129-137.

Wigdor, D., Balakrishnan, R. (2003). *TiltText*: using tilt for text input to mobile phones. In *Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology (UIST'03)*, Vancouver, Canada, November 02-05, ACM Press, New York, NY, 81-90.

Wigdor, D., Balakrishnan, R. (2004). A comparison of consecutive and concurrent input text entry techniques for mobile phones. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI2004)*, ACM Press, New York, NY, USA, 81-88.

Williamson, J., Murray-Smith, R. (2005). Hex: Dynamics and probabilistic text entry. In *Proceedings of the Hamilton Summer School on Switching and Learning in Feedback systems*, R. Murray-Smith, R. Shorten (Eds.), Springer-Verlag, LNCS, Vol. 3355, 333-342.

Wilson, A.D., Agrawala, M. (2006). Text entry using a dual joystick game controller. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'06)*, Montréal, Québec, Canada, April 22-27, ACM Press, New York, NY, 475-478.

Wobbrock, J., Myers, B., Kembel, J. (2003). EdgeWrite : A Stylus-Based Text Entry Method Designed for High Accuracy and Stability of Motion. In *Proceeding of the 16th*

Annual ACM Symposium on User Interface Software and Technology (UIST 2003), ACM Press, New York, NY, USA, 61-70.

Wobbrock, J.O., Myers, B.A. (2005). Gestural text entry on multiple devices. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '05)*, Baltimore, Maryland (October 9-12, 2005), New York: ACM Press, 184-185.

Wobbrock, J.O. (2006). EdgeWrite: A versatile design for text entry and control. *Technical Report CMU-HCII-06-104*, Carnegie Mellon University, July. Disponible à : <http://reports-archive.adm.cs.cmu.edu/anon/hcii/abstracts/06-104.html>

Wobbrock, J.O., Myers, B.A., Chau, D.H. (2006). In-stroke word completion. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '06)*, Montreux, Switzerland (October 15-18, 2006), New York: ACM Press, 333-336.

XPeRT (2003). <http://www.xpertkeyboard.com/>

Zhai, S., Hunter, M., Smith, B.A. (2000). The Metropolis Keyboard - An Exploration of Quantitative Techniques for Virtual Keyboard Design. In *Proceeding of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST 2000)*, San Diego, California, 119-128.

Zhai, S., Hunter, M., Smith, B.A. (2002). Performance Optimization of Virtual Keyboards. *Human-Computer Interaction*, Vol. 17, No 2 & 3, 229-269.

Zhai, S., Kristensson, P-O. (2003). Shorthand Writing on Stylus Keyboard. In *Proceedings of the SIGCHI Conference on Human factors in computing systems (CHI2003)*, ACM Press, New York, NY, USA, 97-104.