



Saisie de texte en contexte dégradé sur interfaces réduites pour dispositifs mobiles

Mohammed Belatar

► To cite this version:

Mohammed Belatar. Saisie de texte en contexte dégradé sur interfaces réduites pour dispositifs mobiles. Interface homme-machine [cs.HC]. Université de Bretagne Sud; Université Européenne de Bretagne, 2009. Français. NNT : . tel-00517822

HAL Id: tel-00517822

<https://theses.hal.science/tel-00517822>

Submitted on 15 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THESE
sous le sceau de : l'Université Européenne de Bretagne

pour obtenir le titre de :
DOCTEUR DE L'UNIVERSITÉ DE BRETAGNE-SUD

Mention : Sciences et Technologies de l'Information et de la
Communication / Informatique et Applications

présentée par
Mohammed Belatar

Préparée au laboratoire VALORIA, EA2593

**École Doctorale Santé, Information-Communications,
Mathématiques et Matière**

Thèse soutenue le 19 janvier 2009

devant le jury composé de :

Benoît MARTIN

Maitre de Conférence (HDR), Université de Metz / *rapporteur*

Christophe KOLSKI

Professeur, Université de Valenciennes / *rapporteur*

Jean-Yves ANTOINE

Professeur, Université de Tours / *examinateur*

Jean-Paul DEPARTE

Ingénieur, Centre Mutualiste de Rééducation et de
Réadaptation Fonctionnelle de Kerpape / *invité*

Franck POIRIER

Professeur, Université de Bretagne-Sud / *Directeur de thèse*

Saisie de texte en contexte dégradé
sur interfaces réduites
pour dispositifs mobiles

الأُمِّي ،

الأَبِّي ،

الأَخُوْتِي ،

أَهْدِي أطروحة الدكتوراه هذه.

À ma mère,

À mon père,

À ma sœur,

À mes frères,

Je dédie affectueusement ma thèse.

REMERCIEMENTS

Je tiens à remercier tous les membres du jury pour m'avoir fait l'honneur d'accepter d'évaluer ce travail de thèse.

Je remercie en particulier mes rapporteurs, Benoît Martin et Christophe Kolski pour les remarques concrètes qu'ils m'ont faites sur les premières « moutures » de mon mémoire de thèse.

Je remercie également Jean-Yves Antoine, d'avoir accepté d'être examinateur et président du jury.

Je suis particulièrement reconnaissant envers Jean-Paul Départe pour les remarques qu'il m'a faites au cours de notre première entrevue, et qui m'ont permis d'avancer dans l'adaptation de mon travail pour les personnes handicapées. Je le remercie aussi pour les moyens qu'il a mis à ma disposition pour pouvoir évaluer une partie de mon travail de thèse au Centre Mutualiste de Rééducation et de Réadaptation Fonctionnelles de Kerpape.

Je remercie bien sûr Franck Poirier, d'une part pour les longues discussions de travail que nous avons eues et grâce auxquelles j'ai pu avancer dans mon projet, et d'autre part pour la chaleur de son encadrement. Je le remercie spécialement pour la lucidité des jugements qu'il a portés à de nombreuses reprises sur mes idées et sur mon travail de réalisation. Ces jugements et les conseils assortis m'ont largement aiguillé sur la voie que j'ai empruntée.

Je remercie également mes collègues et mes amis pour le soutien qu'ils m'ont témoigné, dans les moments les plus difficiles de ma thèse.

Finalement, je remercie ma famille, entre autres pour m'avoir soutenu moralement et financièrement tout au long de mes études. En particulier, j'embrasse ma mère et mon père.

RÉSUMÉ

Cette thèse concerne l'entrée de texte sur les systèmes interactifs mobiles. En particulier une méthode appelée UniGlyph a été conçue. Cette méthode se veut indépendante du système utilisé (assistant numérique, smartphone, GPS...) et utilisable aussi bien par des personnes valides que par des personnes handicapées. Le principe d'UniGlyph consiste à minimiser le nombre de touches utilisées pour la saisie de texte afin d'avoir un clavier de petite taille avec des touches raisonnablement dimensionnées tout en assurant une saisie facile, rapide et confortable. Pour faciliter la saisie, UniGlyph se base sur une analogie scripturale aux caractères romains en majuscules avec une décomposition des caractères en formes primitives. Ensuite, pour assurer une bonne vitesse de saisie, UniGlyph offre un bon dimensionnement des touches en accord avec la loi de Fitts. Finalement, pour réduire l'effort moteur de l'utilisateur, UniGlyph permet de saisir du texte avec un seul appui de touche par caractère comme sur un clavier d'ordinateur de bureau. UniGlyph permet de saisir du texte avec seulement trois touches associées à des formes de base et permet une saisie de niveau mot grâce à un système de désambiguïsation lexicale comme, par exemple, le clavier téléphonique T9.

Outre la conception de méthode de saisie UniGlyph, ce mémoire présente deux implémentations de cette méthode sur assistant numérique (PDA). L'une pour les personnes valides et l'autre pour les personnes handicapées. Pour ces dernières, nous avons privilégié de réaliser une solution adaptée aux handicaps moteurs lourds vu le manque de solutions destinées à cette catégorie d'utilisateurs et le grand intérêt de leur rendre accessible les systèmes d'infocommunication mobiles. Les deux implémentations présentées dans ce mémoire ont été évaluées respectivement par des utilisateurs valides au sein du laboratoire VALORIA et par une personne avec le syndrome d'enfermement (Locked-In Syndrome) au Centre Mutualiste de Rééducation et Réadaptation Fonctionnelle de Kerpape. Les résultats des évaluations sont très encourageants et ce nouveau mode d'écriture a suscité un grand intérêt chez les utilisateurs.

ABSTRACT

This thesis is about text entry on mobile interactive systems. In particular, a new input method called UniGlyph was designed. This method is device independent (PDA, Smartphone, GPS ...) and can be used by both able-bodied and people with motor impairments. The principle of UniGlyph is to minimize the number of keys used for typing text in order to have a small keyboard with reasonably sized keys while ensuring easy, fast and comfortable text input.

To facilitate the text input, UniGlyph uses analogy to the Roman capital letters with a decomposition of the characters into basic shapes called “primitives”. Then, to ensure high input rates, UniGlyph provides well sized keys according to Fitts’ law. Finally, to reduce the requested motor effort, UniGlyph allows typing text with only one keystroke per character like a desktop keyboard. Only three keys are used, each one is labeled by two primitives. UniGlyph allows a word-level entry through a lexical disambiguation system, like the T9 text entry on mobile phone keypad.

In addition to the design of UniGlyph, this thesis presents two implementations of this method on digital assistant (PDA). The first is for the able-bodied and the second is for people with disabilities. For the latter, we preferred to provide a solution for users with severe motor disabilities due to the lack of solutions for this type of users and the general interest of making these mobile communication devices accessible to them. The two implementations presented in this thesis were evaluated by able-bodied users in the VALORIA laboratory and by a person with the Locked-In Syndrome in the Functional Rehabilitation Center of Kerpape (Centre Mutualiste de Rééducation et Réadaptation Fonctionnelles de Kerpape). The evaluation results are very interesting and this new way of writing has aroused great interest among users.

SOMMAIRE

SOMMAIRE	1
TABLES DES ILLUSTRATIONS.....	5
INTRODUCTION	9
I. LA SAISIE DE TEXTE SUR DISPOSITIFS MOBILES	13
1. INTRODUCTION	13
2. MÉTRIQUES D'ÉVALUATION DE LA SAISIE DE TEXTE	14
2.1. <i>Métriques liées à la vitesse : CPS et WPM</i>	14
2.1.1. Les lois de Fitts et de Hick-Hyman	15
2.1.2. Le Keystroke-Level Model.....	18
2.2. <i>Métriques liées à l'effort</i>	20
2.2.1. Le nombre de clics/gestes	20
2.2.2. Les taux d'erreurs	21
2.3. <i>Autre métriques</i>	23
3. ÉTAT DE L'ART DES MÉTHODES DE SAISIE EN MOBILITÉ.....	24
3.1. <i>Méthodes à touches physiques</i>	25
3.1.1. Le clavier téléphonique T9.....	25
3.1.2. Le clavier Q12	26
3.1.3. CyKey	27
3.1.4. EQx.....	28
3.2. <i>Méthodes à interface tactile</i>	29
3.2.1. Les claviers logiciels complets.....	29
3.2.2. Écriture manuscrite	35
3.2.3. ShapeWriter.....	35
3.2.4. Claviature.....	36
3.2.5. PNLH Multitap	37
3.2.6. VirHKey	38
3.2.7. TUP	39
3.2.8. Circumscript.....	39
3.2.9. EasyStroke	40
3.2.10. FreePad.....	41
3.2.11. 7SI	41
3.3. <i>Méthodes à détection de l'inclinaison</i>	42
3.3.1. Unigesture	42
3.3.2. TiltType	43
3.3.3. TiltText	43
3.3.4. Hex.....	43
3.3.5. UniGest	44
4. LA SAISIE DE TEXTE POUR PERSONNES HANDICAPÉES	45
4.1. <i>Méthodes pour handicap visuel</i>	45
4.2. <i>Méthodes pour handicap moteur</i>	46
4.2.1. Saisie analogique	46
4.2.2. Saisie par désignation de cibles	49
4.3. <i>Systèmes de communication palliative</i>	51
5. CONCLUSION	53

II. LA SAISIE ANALOGIQUE À PRIMITIVES	55
1. INTRODUCTION	55
2. MÉTHODES EXISTANTES	55
2.1. <i>Symbol Creator</i>	56
2.2. <i>HandyScript</i>	57
2.3. <i>Neopad</i>	58
3. GLYPH, UNE SAISIE À SIX PRIMITIVES	59
4. GLYPH2, L'AMÉLIORATION	61
5. GLYPH2 POUR POCKET PC	63
6. CONCLUSION	66
III. UNIGLYPH : UNE MÉTHODE DE SAISIE « OPTIMALE ».....	67
1. INTRODUCTION	67
2. UNE MÉTHODE « OPTIMALE »	68
2.1. <i>Règles de conception</i>	68
2.2. <i>Mise en œuvre</i>	69
3. ÉTUDES AU PRÉALABLE	72
3.1. <i>Étude statistique</i>	72
3.2. <i>Évaluation expérimentale</i>	74
4. LE CLAVIER UNIGLYPH SUR PDA	80
4.1. <i>Description</i>	80
4.1.1. L'interface et l'utilisation du clavier.....	80
4.1.1.1. Les éléments de l'interface	80
4.1.1.2. Les modes du clavier	82
4.1.1.3. La configuration du clavier	86
4.1.2. L'architecture logicielle du clavier	90
4.1.2.1. L'architecture générale	90
4.1.2.2. La désambiguïsation et la prédiction	92
4.2. <i>Évaluations</i>	97
4.2.1. Prédiction théorique des performances	97
4.2.2. Évaluation utilisateur	99
4.2.2.1. Protocole de l'évaluation	99
4.2.2.2. Résultats et discussions	103
5. AUTRES IMPLÉMENTATIONS.....	109
6. CONCLUSION	109
IV. HANDIGLYPH : L'ADAPTATION D'UNIGLYPH POUR HANDICAP MOTEUR LOURD	111
1. INTRODUCTION	111
2. PRINCIPES ET INTERFACES.....	112
2.1. <i>Principes</i>	112
2.2. <i>Interface du clavier</i>	113
2.3. <i>Interface de configuration</i>	115
3. ÉVALUATION D'HANDIGLYPH AVEC UNE PERSONNE HANDICAPÉE	118
3.1. <i>Matériel et application expérimentale</i>	118
3.2. <i>Protocole de l'expérience</i>	119
3.3. <i>Résultats et discussions</i>	120
3.3.1. L'effort	120
3.3.2. Le taux d'erreurs.....	122
3.3.3. Les données relatives au temps.....	122
3.3.4. La satisfaction de l'utilisateur	124

4. CONCLUSION	125
CONCLUSIONS ET PERSPECTIVES.....	127
RÉFÉRENCES.....	131
ANNEXES	143
ANNEXE 1 : TABLE DES CARACTÈRES EDGEWRITE	145
ANNEXE 2 : TABLES DES CARACTÈRES GLYPH	146
ANNEXE 3 : RÉSULTATS STATISTIQUES DES DIFFÉRENTES COMBINAISONS DE PRIMITIVES SUR LES TOUCHES.	147
ANNEXE 4 : STRUCTURE GLOBALE DE UNIGLYPHLIB	148
ANNEXE 5 : STRUCTURE GLOBALE DE HANDIGLYPHLIB	149
ANNEXE 6 : LISTE DES PHRASES UTILISÉES DANS LES ÉVALUATIONS	150
ANNEXE 7 : FICHE DE PRÉSENTATION DES ÉVALUATIONS	153
ANNEXE 8 : FICHE DE PRÉSENTATION DU CLAVIER UNIGLYPH	155
ANNEXE 9 : FICHE DE PRÉSENTATION DU CLAVIER PHRAZE-IT	157
ANNEXE 10 : FICHE DE PRÉSENTATION DU CLAVIER 4-KEY EDGEWRITE.....	159
ANNEXE 11 : FICHE DE PRÉSENTATION DE HANDIGLYPH	161

TABLES DES ILLUSTRATIONS

TABLE DES FIGURES

FIGURE I-1. LES EXPÉRIENCES MENÉES PAR FITTS ET QUI ONT DONNÉ NAISSANCE À SA LOI.	15
FIGURE I-2. LA PREMIÈRE EXPÉRIENCE MENÉE PAR HICK QUI A DONNÉ NAISSANCE À LA LOI DE HICK-HYMAN APRÈS LES TRAVAUX DE HYMAN.	17
FIGURE I-3. ALGORITHME POUR CALCULER LA DISTANCE DE LEVENSSTEIN. EXEMPLE AVEC LES DEUX CHAÎNES DE CARACTÈRES « ABCD » ET « ACBD ».	22
FIGURE I-5. LE CLAVIER Q12 CLASSIC AVEC SON ORGANISATION ALPHABÉTIQUE ET LE CLAVIER Q12 COMPUTER AVEC SON ORGANISATION QWERTY.	27
FIGURE I-6. LE CLAVIER CYKEY, SON ANCÊTRE ET SA TABLE DE CARACTÈRES.	28
FIGURE I-7. EXEMPLES DE CLAVIERS EQX.	29
FIGURE I-8. EXEMPLES DE CLAVIERS LOGICIELS OPTIMISÉS PAR AGENCEMENT.	34
FIGURE I-9. L'INTERFACE DE CLAVIATURE. EXEMPLE DE LA SAISIE DE LA LETTRE 'A'.	37
FIGURE I-10. L'INTERFACE DU PROTOTYPE DU CLAVIER MULTITAP PNLH.	37
FIGURE I-11. L'ALPHABET VIRHKEY. LE CARACTÈRE ESPACE CORRESPOND À UN SIMPLE CLIC, LES VOYELLES CORRESPONDENT À UN GESTE SIMPLE ET LES AUTRES LETTRES CORRESPONDENT À DES GESTES COMPOSÉS. CERTAINES LETTRES SONT ACCESSIBLES PAR DIFFÉRENTS CHEMINS.	38
FIGURE I-12. EXEMPLE DE LA SAISIE DE LA LETTRE 'Q' AVEC LA MÉTHODE VIRHKEY.	38
FIGURE I-13. ORGANISATION DES CARACTÈRES AUTOUR DE LA MOLETTE TACTILE.	39
FIGURE I-14. LA MÉTHODE DE SAISIE CIRCUMSCRIPT. EXEMPLE DE LA SAISIE DU MOT « BONJOUR ».	40
FIGURE I-15. LA LISTE DES PRIMITIVES ET LA CARTE DES CARACTÈRES EASYSTROKE.	40
FIGURE I-16. LE PRINCIPE DE L'ÉCRITURE AVEC FREEPAD.	41
FIGURE I-17. L'ALPHABET 7SI.	41
FIGURE I-18. MÉTHODES DE SAISIE PAR DÉTECTION DE L'INCLINAISON.	42
FIGURE I-19. L'INTERFACE DE LA MÉTHODE DE SAISIE HEX. UNE GRILLE D'HEXAGONES ÉTENDUE À L'INFINI.	44
FIGURE I-20. L'ALPHABET UNIGEST ET LE JEU DE PRIMITIVES ASSOCIÉ.	44
FIGURE I-21. CARTE DE CARACTÈRES SIMPLIFIÉE ET DIFFÉRENTES IMPLÉMENTATIONS D'EDGEWRITE.	48
FIGURE I-22. WEEGIE : EXEMPLES DE SAISIE DES LETTRES 'A', 'B', 'C' ET 'H' (RESPECTIVEMENT DE GAUCHE À DROITE).	49
FIGURE I-23. DASHER, LES DERNIÈRES LETTRES SÉLECTIONNÉES SONT 'R', 'E' ET ESPACE.	50
FIGURE I-24. HEX-O-SPell, SAISIE AVEC UNE INTERFACE CERVEAU-ORDINATEUR. EXEMPLE DE LA SAISIE DE LA LETTRE 'N'.	50
FIGURE I-25. KeCOM, POUR LA SAISIE DE TEXTE À UN SEUL BOUTON POUSSOIR SUR PDA.	53
FIGURE II-1. SYMBOL CREATOR, UNE MÉTHODE DE SAISIE ANALOGIQUE PAR DÉCOMPOSITION DES CARACTÈRES.	56
FIGURE II-2. HANDYSCRIPT, UNE MÉTHODE ANALOGIQUE À ONZE PRIMITIVES.	57
FIGURE II-3. NEOPAD, UNE MÉTHODE ANALOGIQUE À DIX PRIMITIVES.	58
FIGURE II-4. TABLE DES CARACTÈRES ALPHABÉTIQUES ET NUMÉRIQUES GLYPH.	61
FIGURE II-5. DIFFÉRENTES IMPLÉMENTATIONS POSSIBLES DE GLYPH.	61
FIGURE II-6. LE CODAGE DES CARACTÈRES ALPHABÉTIQUES ET NUMÉRIQUE AVEC GLYPH2.	62
FIGURE II-7. L'INTERFACE PRINCIPALE DE GLYPH2PPC.	64

FIGURE II-8. EXEMPLE DE LA SAISIE DE LA LETTRE 'X' AVEC LES DIFFÉRENTS MODES DE SAISIE DE GLYPH2PPC.	64
FIGURE II-9. ACCÈS AUX DIFFÉRENTS MODES ET FONCTIONS VIA LE MODE DE COMMANDE.	65
FIGURE III-1. SCHÉMA DU PRINCIPE ET DES BASES DE LA CONCEPTION D'UNIGLYPH.....	70
FIGURE III-2. ANALOGIE AUX CARACTÈRES MANUSCRITS EN MINUSCULES.	74
FIGURE III-3. ANALOGIE AUX CARACTÈRES IMPRIMÉS EN MAJUSCULES.	74
FIGURE III-4. FRÉQUENCES DES LETTRES DANS LES 22 MOTS DE NOTRE CORPUS PAR RAPPORT AUX FRÉQUENCES DES LETTRES DANS LA LANGUE FRANÇAISE.	75
FIGURE III-5. APPLLET D'ÉVALUATION DE L'ANALOGIE MINUSCULE. EXEMPLE DE LA SAISIE DU MOT « RAPPELER » AVEC UNE ERREUR SUR LA LETTRE 'L'	76
FIGURE III-6. APPLLET D'ÉVALUATION DE L'ANALOGIE MAJUSCULE. EXEMPLE DE LA SAISIE DU MOT « REMPLIR ».	76
FIGURE III-7. COMPARAISON DES TAUX DE CORRECTIONS PAR LETTRE AVEC LES DEUX SOLUTIONS PROPOSÉES.....	78
FIGURE III-8. TAUX D'ERREURS SUR LES TROIS PREMIÈRES SESSIONS.	79
FIGURE III-9. INTERFACE PRINCIPALE DU CLAVIER UNIGLYPH SUR PDA.	81
FIGURE III-10. LE SYMBOLE DE LA TOUCHE DE COMMANDE SUR DIFFÉRENTES MÉTHODES DE SAISIE.	81
FIGURE III-11. LES DIFFÉRENTS MODES DU CLAVIER UNIGLYPH.	82
FIGURE III-12. LE FEEDBACK LORS DE LA SAISIE. EXEMPLE DE LA SAISIE DE LA PHRASE « J'AI DIT BONJOUR », LE MOT EN COURS DE SAISIE EST « BONJOUR », L'UTILISATEUR A DÉJÀ EFFECTUÉ 5 APPUIS SUR LES TOUCHES DE PRIMITIVES, IL LUI RESTE À EFFECTUER LES APPUIS CORRESPONDANTS AUX LETTRES 'U' ET 'R'.	84
FIGURE III-13. FENÊTRE D'AJOUT DE NOUVEAUX MOTS DANS LE DICTIONNAIRE. EXEMPLE DE L'ENTRÉE « MOHAMMED ».	85
FIGURE III-14. REPRÉSENTATION DES CHIFFRES EN UNE À DEUX PRIMITIVES.	85
FIGURE III-15. REPRÉSENTATION DES CHIFFRES EN UNE SEULE PRIMITIVE.	86
FIGURE III-16. L'OPTION DES COMMANDES GESTUELLES.	88
FIGURE III-17. LE MODE DE COMMANDE ACCÉDÉ PAR UN APPUI LONG SUR LA TOUCHE DE COMMANDE.	88
FIGURE III-18. UTILISATION DES TOUCHES PHYSIQUES DU PAVÉ DIRECTIONNEL.	89
FIGURE III-19. AFFICHAGE DES LETTRES SUR LES TOUCHES DE PRIMITIVES.	89
FIGURE III-20. ARCHITECTURE GLOBALE DE LA DLL UNIGLYPHLIB.	91
FIGURE III-21. REPRÉSENTATION DU DICTIONNAIRE EN ARBRE DE SEGMENTS.	94
FIGURE III-22. LES DIMENSIONS DU CLAVIER UNIGLYPH.	98
FIGURE III-23. LES DEUX CLAVIERS ÉVALUÉS AVEC UNIGLYPH.	100
FIGURE III-24. PROTOTYPES DE L'ÉVALUATION.	102
FIGURE III-25. NOMBRE MAXIMAL DE MOTS SAISIS PAR MINUTE (WPM) PAR PARTICIPANT.	103
FIGURE III-26. NOMBRE MOYEN DE MOTS SAISIS PAR MINUTE (WPM) PAR SESSION.....	104
FIGURE III-27. NOMBRE MOYEN D'APPUIS PAR CARACTÈRE (KSPC) PAR SESSION.	105
FIGURE III-28. TAUX D'ERREURS MOYEN OBTENU AVEC CHACUNE DES TROIS MÉTHODES.....	106
FIGURE III-29. AVIS DES PARTICIPANTS SUR LES TROIS MÉTHODES À L'ISSUE DE LA CINQUIÈME SESSION.	108
FIGURE III-30. ÉVOLUTION DE LA NOTATION DE CHAQUE MÉTHODE AU LONG DE L'EXPÉRIENCE.	108
FIGURE III-31. EXEMPLES DE PDA, DE TÉLÉPHONE MOBILE ET DE SMARTPHONE ÉQUIPÉS DE DIFFÉRENTS TYPES DE MOLETTES QUI PEUVENT ÊTRE EXPLOITÉES PAR UNIGLYPH.....	109
FIGURE IV-1. LE CLAVIER HANDIGLYPH EST COMPOSÉ DE 4 TOUCHES ET DE DEUX ZONES D'AFFICHAGE ET DE SÉLECTION.	113
FIGURE IV-2. DANS LE MODE DE COMMANDE, LES TOUCHES DE PRIMITIVES DEVIENNENT DES TOUCHES DE FONCTIONS ET LA ZAS DU MILIEU PERMET D'ACCÉDER AUX AUTRES MODES.	114
FIGURE IV-3. DANS LA FENÊTRE DE CONFIGURATION DE HANDIGLYPH, LES DIFFÉRENTES OPTIONS SONT BALAYÉES EN CONTINUE. ..	115
FIGURE IV-4. LA SOURIS BLUETOOTH ÉQUIPÉE D'UN CONNECTEUR POUR BRANCHER LE CONTACTEUR UTILISÉ PAR LE TESTEUR.	119

FIGURE IV-5. COPIE D'ÉCRAN DU PROTOTYPE EXPÉRIMENTAL.	120
FIGURE IV-6. NOMBRE D'APPUIS PAR CARACTÈRE SUR CINQ SESSIONS.	121
FIGURE IV-7. LES TAUX D'ERREURS RÉSIDENTIELLES DANS LE TEXTE TRANSCRIT.	122
FIGURE IV-8. ÉVOLUTION DE LA VITESSE DU BALAYAGE.	122
FIGURE IV-9. POURCENTAGES DU TEMPS PASSÉ SUR LES ZAS PAR RAPPORT AU TEMPS TOTAL DE LA TÂCHE DE SAISIE.	123
FIGURE IV-10. POURCENTAGES DES MOTS TROUVÉS DANS CHAQUE LISTE.	123

TABLE DES TABLEAUX

TABLEAU I-1. AUTRES MÉTRIQUES UTILISÉES POUR L'ÉVALUATION DES MÉTHODES DE SAISIE DE TEXTE.	24
TABLEAU I-2. EXEMPLES DE CLAVIERS LOGICIELS OPTIMISÉS PAR AGENCEMENT.	33
TABLEAU I-3. COMPARAISON DES MÉTHODES SHAPEWRITER, UNISTROKES ET GRAFFITI EN TERMES DE « GESTURE PRODUCTION TIME ».....	36
TABLEAU II-1. LES SIX PRIMITIVES DE FORMES DANS LA DÉCOMPOSITION GLYPH.	60
TABLEAU II-2. DÉCOMPOSITION DES LETTRES 'G' ET 'H' EN PRIMITIVES GLYPH.	60
TABLEAU II-3. LES PHRASES UTILISÉES POUR L'ÉVALUATION DU CLAVIER GLYPH2PPC.	65
TABLEAU III-1. EXEMPLES DE JEUX DE PRIMITIVES ET DE TOUCHES ÉTUDIÉS.	73
TABLEAU III-2. LISTE DES MOTS UTILISÉS DANS L'ÉVALUATION PRÉLIMINAIRE EN LIGNE.	74
TABLEAU III-3. RÉSUMÉ DES PROFILS DES PARTICIPANTS.	77
TABLEAU III-4. RÉPARTITION DES PARTICIPATIONS PAR APPLLET ET PAR SESSION.	77
TABLEAU III-5. LES COMMENTAIRES DES PARTICIPANTS (AUTHENTIQUEMENT RECOPIÉS).	79
TABLEAU III-6. STRUCTURE DE L'INTERFACE DE COMMUNICATION DE L'APPLICATION AVEC LE CLAVIER.	91
TABLEAU III-7. RÉSULTATS DE LA SIMULATION DES TROIS SYSTÈMES DE PRÉDICTION.	97
TABLEAU III-8. PROBABILITÉS DES SUCCESSIONS DE TOUCHES.	98
TABLEAU III-9. EXEMPLES DE PHRASES DU CORPUS DE L'ÉVALUATION.	101
TABLEAU III-10. LE FORMULAIRE D'ÉVALUATION DE LA SATISFACTION DES UTILISATEURS.	107
TABLEAU III-11. RÉSULTATS DES QUESTIONNAIRES DE SATISFACTION.	107
TABLEAU IV-1. LE FORMULAIRE D'ÉVALUATION DU NIVEAU DE SATISFACTION DE L'UTILISATEUR.	124

INTRODUCTION

Le nombre de SMS échangés ne cesse d'augmenter avec l'explosion du nombre de téléphones mobiles. En 2002, la distribution du nombre de téléphones mobiles sur la population mondiale était de 59,3 par 100 personnes. 360 milliards de SMS ont été échangés dans le monde durant cette année avec une moyenne mensuelle de 30 milliards de SMS. Ces chiffres évoluent très rapidement avec l'évolution démographique, la progression de la technologie et l'accessibilité des prix. Le nombre d'abonnés à la téléphonie mobile est estimé à 3,25 milliards en 2007 ce qui correspond à 50% de la population mondiale avec une croissance annuelle de 25% constatée entre 2006 et 2007. Pour l'année 2008, l'Union Internationale des Télécommunications¹ prévoit plus de 4 milliards d'abonnés à la téléphonie mobile. Le nombre de SMS échangés dans le monde est estimé à 2 300 milliards cette année².

Les SMS ne représentent qu'une petite partie du texte que saisit un utilisateur sur un dispositif d'infocommunication mobile. Ces dernières années, l'internet mobile a eu une progression phénoménale avec l'explosion des téléphones « intelligents » (Smartphones) et de la diversification de la connectivité en technologies 3G, GPRS et WiFi. Selon l'Association Française du Multimédia Mobile³, 25% des possesseurs de téléphones mobiles en France âgés entre 15 et 50 ans sont des internautes mobiles (mobinautes) et 41,1% des possesseurs d'un téléphone mobile ont désormais la possibilité d'accéder à Internet. Cela conduit à de nouvelles utilisations du téléphone mobile sollicitant beaucoup plus la saisie de texte comme la navigation sur Internet (remplissage de formulaire, saisie d'adresses web...), l'envoi de courriels et la discussion instantanée.

La saisie de texte est donc devenue un des grands challenges de la téléphonie mobile. De nombreuses méthodes de saisie de texte ont été développées ces dernières années. Ces nouvelles méthodes cherchent à réduire la pénibilité et le nombre d'erreurs, à augmenter la vitesse de la saisie et à libérer plus d'espace sur, ou pour l'écran de l'appareil mobile (respectivement dans le cas d'un clavier logiciel et dans le cas d'un clavier physique). Pour la plupart, ceux-ci restent toutefois fort mal adaptés aux petites dimensions des téléphones mobiles, surtout lorsque les concepteurs reviennent au clavier classique AZERTY (ou QWERTY) pour faciliter la saisie (voir les exemples dans la figure qui suit).

¹ Union International des Télécommunications, site web : www.itu.int/

² Agence de recherche et de consultation en technologies d'information, www.gartner.com/

³ Association Française du Multimédia Mobile, « Résultats étude AFFM : les services mobiles », www.pro.gallery.fr/



Les téléphones mobiles avec toutes leurs déclinaisons (téléphone intelligent, PDA phone, ...) ne sont pas les seuls dispositifs mobiles qui rencontrent cette problématique de la saisie de données. Plusieurs autres types de dispositifs mobiles ont émergé ces dix dernières années. La figure ci-dessous montre quelques exemples de dispositifs mobiles qui requièrent une entrée de texte.



Dans ce contexte, nous étudions dans cette thèse la possibilité de concevoir une méthode de saisie de texte indépendante du dispositif selon un principe « learn once, write everywhere » [Wobbrock et al., 2005] tout en respectant tous les critères et les limitations imposées par ce genre de dispositifs mobiles à petites dimensions et à faibles ressources.

Un autre aspect aussi important que nous traitons dans cette thèse est l'accessibilité de ces dispositifs aux personnes handicapées. En effet, ces dispositifs électroniques peuvent leur être d'une grande utilité car ils leur offrent une plus grande autonomie que ne le ferait un ordinateur de bureau. Nous nous concentrons surtout sur les personnes lourdement handicapées ayant perdu l'usage de la parole tout en ayant perdu la plupart de leurs contrôles moteurs (tétraplégiques, locked-in syndromes, infirmes moteurs cérébraux...). Jusqu'à maintenant, très peu de solutions ont été proposées à cette catégorie d'utilisateurs.

Ce mémoire commence donc par une première partie où nous effectuons un examen attentif des méthodes de saisie déjà existantes pour relever les problèmes de chacune et le manque de solutions de saisie de texte pour les personnes ayant un handicap moteur lourd. Ensuite, dans un deuxième chapitre, nous nous penchons sur les méthodes de saisie analogiques consistant à décomposer les caractères en formes de base et principalement sur la méthode Glyph développée dans notre laboratoire et finalisée au début de cette thèse. Les deux chapitres suivants détaillent notre contribution, le

premier concerne l'analyse et la conception d'une nouvelle méthode de saisie ainsi que son implémentation sur un dispositif mobile et son évaluation avec des personnes valides, le dernier chapitre porte sur une adaptation de cette méthode pour les personnes handicapées.

CHAPITRE I

LA SAISIE DE TEXTE SUR DISPOSITIFS MOBILES

Résumé

Ce chapitre constitue un état de l'art des entrées de texte sur les dispositifs électroniques mobiles comme les téléphones et les assistants numériques. Vu le grand nombre de méthodes de saisie existantes, nous nous focalisons sur les méthodes les plus récentes tout en étudiant certaines des méthodes les plus classiques. Le chapitre commence par une étude des métriques les plus utilisées pour l'évaluation de ces méthodes et se termine par une étude des méthodes destinées aux personnes handicapées.

1. Introduction

Un grand nombre de difficultés relatives à l'entrée de texte ont émergé avec le développement de l'informatique mobile. La majorité de ces problèmes est due aux dimensions de plus en plus réduites de ces dispositifs et aux conditions instables de leur utilisation. Si un ordinateur personnel traditionnel ne peut être utilisé que sur un bureau, un ordinateur portable est fréquemment utilisé sur les genoux, par exemple dans les transports. Un ordinateur de poche quant à lui, comme un téléphone mobile, peut être utilisé dans des conditions encore plus variées (assis, debout, en marchant ...). Le clavier AZERTY (physique ou virtuel) n'est plus adapté à ces nouveaux modes d'utilisation à cause de son nombre important de touches qui induit à une surface de grande dimension. Dans un premier temps, des techniques de saisie basées sur la reconnaissance de caractères ou de formes ont été développées (Block Recognizer, Transcriber, Graffiti® ...). Ces techniques sont généralement lentes et présentent des taux d'erreur importants [Sears et Arora, 2002 ; Bouteruche et al., 2005] dus à la variabilité de l'écriture et aux performances limitées des algorithmes de reconnaissance. Ensuite, d'autres méthodes de saisie ont été développées pour répondre aux exigences de la saisie en mobilité mais aucune n'est actuellement totalement satisfaisante.

Pour donner au lecteur une idée de la multitude des méthodes qui ont été élaborées au fil des dernières années et de leurs performances, nous en présentons brièvement quelques unes dans ce chapitre que nous avons choisies soit parce qu'elles sont nouvelles et donc peu ou non référencées dans les ouvrages et les thèses déjà publiées, soit parce qu'elles constituent des références incontournables dans le domaine. Nous commençons par étudier les principales métriques et techniques d'évaluation des méthodes de saisie de texte. Ensuite, nous décrivons brièvement les méthodes sélectionnées en indiquant

leurs points forts et leurs faiblesses. À la fin du chapitre nous étudions les méthodes de saisie sur dispositifs mobiles adaptées aux utilisateurs handicapés.

2. Métriques d'évaluation de la saisie de texte

Les dispositifs de saisie de texte représentent une partie de l'interface humain-machine. La plupart des règles de conception et recommandations ergonomiques en IHM sont applicables pour l'entrée de texte, mais il est important de considérer les métriques qui sont spécifiques ou plus adaptées à l'entrée de texte. Nous considérons ici les principales métriques qui permettent de quantifier les performances d'une méthode de saisie et classifions ces métriques en deux catégories selon la nature de l'élément de performance quantifié.

2.1. Métriques liées à la vitesse : CPS et WPM

Le CPS (Character Per Second) permet généralement de quantifier la vitesse d'une entrée ou d'une sortie de type textuel comme les imprimantes matricielles ou à marguerite par exemple. 1 CPS est approximativement l'équivalent de 10 bits par seconde (bit/s) dans les communications asynchrones et 8 bit/s dans les cas de communications synchrones [Grenier, 2000] vu qu'un caractère peut être codé en 8 bits et que la communication asynchrone nécessite des informations supplémentaires pour délimiter l'information transmise.

Dans le domaine de la saisie de texte, le CPS désigne le nombre de caractères saisis par seconde. Il est calculé en divisant la longueur du texte entré par la durée de la saisie en seconde comme le montre l'équation ci-dessous où S (string) correspond au texte saisi et T au temps de saisie.

$$CPS = \frac{|S|}{T} \quad (1)$$

Le CPS est souvent un petit nombre décimal dont au moins les dixièmes et les centièmes sont souvent significatifs. Pour cela, une autre métrique, le WPM (Words Per Minute), est souvent préférée pour faciliter les comparaisons. Elle permet d'exprimer la vitesse de saisie en termes de nombre de mots entrés par minute. Le WPM est la mesure empirique de performances la plus utilisée dans les travaux sur les entrées de texte. Un mot est défini dans ce contexte comme étant un ensemble de 5 caractères qui peuvent inclure des lettres, des chiffres, des espacements ou tout autre type de caractères. Le WPM est calculé selon l'équation suivante :

$$WPM = \frac{|S| \times 60}{T \times 5} = CPS \times \frac{60}{5} \quad (2)$$

Pour plus de précision dans l'évaluation empirique de la vitesse de saisie, le premier caractère est souvent exclu du calcul. Dans ce cas, le chronométrage de la saisie est déclenché par la saisie du premier caractère et la longueur du texte S considérée est égale à la longueur du texte transcrit moins 1.

Le CPS et le WPM peuvent aussi être approximativement prédites pour une méthode donnée dans un contexte donnée. Dans la suite, nous décrivons les deux modèles prédictifs les plus utilisés.

2.1.1. Les lois de Fitts et de Hick-Hyman

En se basant sur le théorème de Shannon publié à la fin des années 40 [Shannon, 1948], Fitts a fait plusieurs expériences sur le mouvement entre deux cibles en variant la largeur de ces cibles, la distance entre elles et le dispositif de pointage (voir Figure I-1). Dans l'une de ces expériences, les sujets déplaçaient un pointeur en va-et-vient entre deux barres métalliques le plus rapidement possible tout en tapant au centre de chacune [Fitts, 1954]. Ce dispositif expérimental est communément appelé le « Fitts' paradigm ». Dans d'autres expériences, les sujets ont été invités à déplacer des petits disques ou des épinglettes.

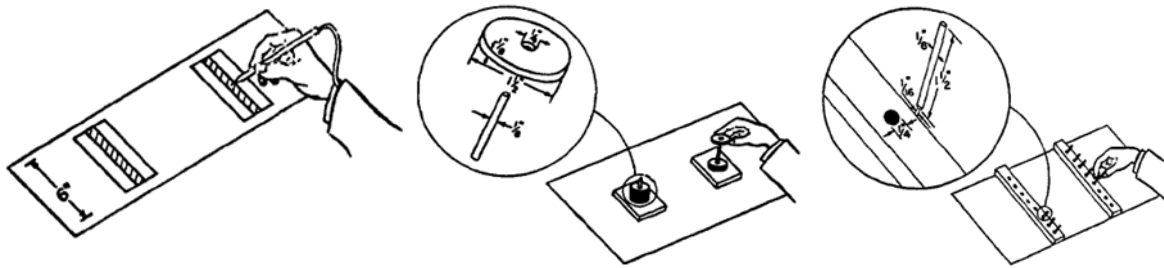


Figure I-1. Les expériences menées par Fitts et qui ont donné naissance à sa loi.

Les résultats obtenus ont permis de mettre en relation l'amplitude du mouvement, la largeur de la cible et le temps nécessaire pour effectuer le mouvement vers le centre de cette cible comme le montrent les équations suivantes :

$$MT = \frac{ID}{IP} \quad (3)$$

$$ID = \log_2 \left(\frac{2A}{W} \right) \quad (4)$$

Où :

- IP est l'index de performance, en d'autres termes la capacité du système moteur humain,
- ID est l'index de difficulté de la tâche motrice, défini par Fitts,
- MT est le temps de mouvement d'un point de départ jusqu'au centre de la cible,
- A est l'amplitude du mouvement,
- W est la largeur de la cible (dans le sens du mouvement).

Dans les décennies qui ont suivi la publication originale de Fitts, sa loi s'est avérée une des plus robustes. Elle est devenue l'une des plus citées, et l'un des modèles les plus largement adoptés parmi ceux qui ont émergé de la psychologie expérimentale. Les études psychomotrices dans divers contextes ont constamment montré une forte corrélation entre l'indice de difficulté de Fitts et le temps effectivement nécessaire pour effectuer une tâche de mouvement (sous certaines conditions).

Le domaine de l'Interaction Humain-Machine est particulièrement riche d'enquêtes sur la performance humaine utilisant la loi de Fitts. Plusieurs travaux ont été menés pour adapter cette loi aux différents types de tâches d'interaction et plusieurs redéfinitions de l'index de difficulté ont été proposées. On note essentiellement les recherches menées par MacKenzie durant sa thèse de doctorat qu'il a soutenue en 1991 et dont le sujet a porté principalement sur la loi de Fitts. Les résultats de ses recherches

ont été publiés plus tard dans [MacKenzie, 1992] et [MacKenzie et Buxton, 1992] où il étudie l'application de cette loi en deux dimensions, ou encore dans [MacKenzie, 1995] et [Soukoreff et MacKenzie, 1995] où il explique en détail l'application de la loi de Fitts pour la prédiction de la vitesse de saisie avec un stylet sur un clavier logiciel. La loi de Fitts adaptée prend la forme suivante :

$$MT_{ij} = \begin{cases} TM_{rep} , & i = j \\ a + b \times \log_2 \left(\frac{A_{ij}}{W_j} + 1 \right), & i \neq j \end{cases} \quad (5)$$

Avec :

- MT_{ij} est le temps de mouvement entre une touche i et une touche j ,
- TM_{rep} est le temps moyen pour re cliquer sur la même touche,
- a et b sont des constantes,
- A_{ij} est la distance (amplitude) entre le centre de la touche i et celui de la touche j ,
- W_j est la largeur de la cible dans le sens du mouvement.

Les constantes a et b peuvent prendre différentes valeurs selon le contexte de l'application de cette loi : le dispositif de pointage, le support et la tâche à effectuer sont les éléments principaux qui affectent ces valeurs. En plus des valeurs obtenues par Fitts lors de ses expériences, plusieurs études ont proposé des valeurs différentes dans des situations différentes. On note notamment les travaux de MacKenzie et ses collègues [MacKenzie et al., 1991] et de Zhai et ses collègues [Zhai et al., 2002b] qui redéfinissent expérimentalement les valeurs des constantes de la loi de Fitts selon le contexte. Dans la suite de ce mémoire, nous utiliserons les valeurs les plus adaptées à notre cas (saisie à stylet avec un clavier logiciel sur dispositif mobile) à savoir : $a = 0$ et $b = 1/4,9$.

Le temps pour re cliquer sur la même touche dépend aussi du dispositif et plusieurs valeurs ont été déduites expérimentalement dans [Soukoreff et MacKenzie, 1995], [MacKenzie et Zhang, 1999] et dans le mémoire de master de Zhang [Zhang, 1998]. Nous avons choisi d'utiliser le TM_{rep} donné par Soukoreff et MacKenzie (153 ms) pour la similitude du contexte de leur expérience avec notre travail.

Dans le cas d'une méthode de saisie à un clic par caractère, le temps de mouvement entre une touche i et une touche j correspond à la saisie du caractère affecté à la touche j . Le MT_{ij} est donc pondéré par la probabilité P_{ij} de succession de la touche i par la touche j ($\sum P_{ij} = 1$) pour calculer le temps de mouvement moyen $\overline{MT}$. Le CPS et le WPM peuvent être ensuite calculés selon les équations suivantes :

$$\overline{MT} = \sum_{i,j \in \text{touches}} (P_{ij} \times MT_{ij}) \quad (6)$$

$$CPS = \frac{1}{(\overline{MT})} \quad (7)$$

$$WPM = \frac{60}{5 \times \overline{MT}} \quad (8)$$

Cette méthode de prédiction de la vitesse de saisie ne repose que sur le temps de mouvement. Elle donne une estimation maximale de la vitesse qui peut être théoriquement atteinte par un utilisateur suffisamment expérimenté pour la méthode étudiée et donc, qui anticipe et succède les clics les uns après les autres sans interruption. Une autre donnée, appelée « temps de réaction » ou « temps de recherche visuelle », est souvent ajoutée au temps de mouvement pour mieux prédire la vitesse de la saisie. Pour ce faire, la loi de Hick-Hyman est appliquée conjointement avec la loi de Fitts.

Dans ces premières expériences en 1952, Hick a essayé de démontrer que le temps de réaction est proportionnel à la quantité d'information manipulée [Hick, 1952]. Il a relié 10 lampes à un dispositif qui en allume aléatoirement une à la fois. Les sujets ont été invités à mettre leurs doigts sur 10 boutons pour appuyer le plus rapidement possible sur celui qui correspond à la lampe qui s'allume (voir Figure I-2).

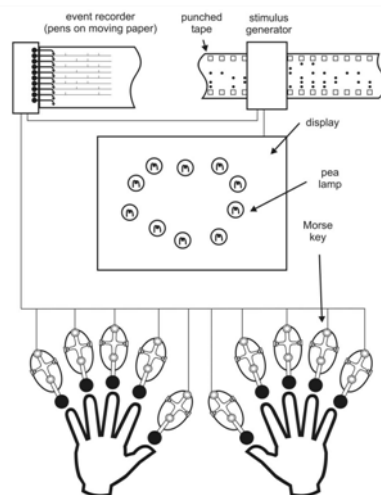


Figure I-2. La première expérience menée par Hick qui a donné naissance à la loi de Hick-Hyman après les travaux de Hyman.

Les résultats des expériences de Hick ont montré que la relation entre le temps de réaction et le nombre d'alternatives (items) est logarithmique. En 1953, Hyman a affiné les expériences et a étendu les résultats de Hick [Hyman, 1953] pour obtenir l'équation ci-dessous. Cette équation n'est pas la même que celle présentée par Hick mais elle confirme bien que le temps de réaction est proportionnel au nombre d'informations transmises.

$$RT = a + b \times \log_2(N) \quad (9)$$

Où :

- RT est le temps de réaction,

- a et b sont des constantes déterminées empiriquement. Contrairement à celles de Fitts, ces constantes n'ont pas été mises en cause depuis les travaux de Hyman et elles ont été utilisées tel quel, à savoir : $a = 0$ et $b = 1/5$ pour une estimation maximale du temps de réaction ou $b=1/7$ pour une estimation minimale de RT.
- N est le nombre d'informations (ou d'items).

Avec l'extension de Hyman, la loi de Hick a été acceptée par beaucoup comme la loi de Hick-Hyman. Son application avec la loi de Fitts à la saisie de texte donne :

$$CPS = \frac{1}{(\overline{MT} + RT)} \quad (10)$$

$$WPM = \frac{60}{5 \times (\overline{MT} + RT)} \quad (11)$$

2.1.2. Le Keystroke-Level Model

Après les lois de Fitts et de Hick-Hyman, Le Kestroke-Level Model (KLM), aussi appelé « modèle Keystroke » dans la littérature, est un autre modèle de prédiction de performances très utilisé dans la conception et l'évaluation des systèmes interactifs. Proposé par Card en 1980, le KLM est une simplification de GOMS (Goals, Operators, Methods, Selection rules). Par rapport à GOMS, KLM permet d'observer seulement le temps d'exécution de la tâche et non le temps total de la réalisation de la tâche. Le temps de réalisation d'une tâche étant la somme du temps d'acquisition et du temps d'exécution. Le temps d'acquisition correspond à la construction d'une représentation mentale de la tâche par l'utilisateur et le temps d'exécution correspond à la réalisation « effective » de la tâche [Diaper et Stanton, 2003]¹. L'autre différence principale avec GOMS est que le modèle KLM ne propose pas de choix de méthode : la méthode est unique et fixée.

En comparaison à d'autres modèles, KLM est relativement facile à appliquer pour la construction et l'évaluation des applications car il se limite seulement aux opérateurs nécessaires pour prédire les performances temporelles de l'utilisateur. Il décompose l'interaction en opérations élémentaires qui sont associées à différents opérateurs. La prédiction du temps minimal nécessaire à un utilisateur expérimenté pour effectuer une tâche spécifique peut alors être facilement déduite par la modélisation des sous-tâches dont elle est composée, en additionnant les temps de chaque opérateur.

KLM introduit six opérateurs pour décrire l'exécution d'une tâche élémentaire. Le temps « d'exécution » d'une tâche est donc la somme des temps correspondants à chaque classe d'opérateurs [Card et al., 1980] :

$$T_{execute} = T_K + T_P + T_H + T_D + T_M + T_R \quad (12)$$

Un ou plusieurs opérateurs peuvent être ignorés selon le type de tâche évaluée. Les différents opérateurs du modèle KLM dans le cas d'une interaction classique de type WIMP sont :

- K pour Keystroke : Cet opérateur correspond à un clic sur une touche. Le T_K est estimé entre 0.08 et 1.2 secondes [Card et al., 1980],

¹ Pages de 83 à 86.

- P pour Pointing : Cet opérateur représente l'action de déplacer la souris pour pointer le curseur à un endroit désiré sur l'écran. Le temps nécessaire est déterminé par la loi de Fitts. Pour les situations typiques, il va de 0,8 à 1,5 s, avec une moyenne de 1,1 s. Cette moyenne est souvent utilisée au lieu de valeurs plus précises si une grande précision n'est pas recherchée.
- H pour Homing : Cet opérateur correspond au repositionnement des mains sur le clavier ou un autre dispositif d'interaction (par exemple : le passage de la main du clavier à la souris ou vis-versa). Le temps correspondant à cet opérateur est généralement de 0,4 s.
- $D(n_D, l_D)$ pour Drawing : Cet opérateur correspond au dessin « manuel » d'une trajectoire de n_D segments linéaires ayant une longueur totale de l_D cm. Le temps correspondant est estimé à $0,9 n_D + 0,16 l_D$.
- M pour Mental : Cet opérateur représente la routine de penser ou de réfléchir lors de la tâche interactive. Bien sûr, le temps nécessaire pour effectuer un acte mental dépend des processus cognitifs impliqués. Il peut être très variable d'une situation à une autre et d'une personne à une autre. Cet opérateur est basé sur le fait que, lorsqu'un utilisateur expérimenté entre en interaction avec le système dans une tâche courante, des pauses de l'ordre d'une seconde sont toujours observés. T_M est estimé entre 0,6 et 1,35 s. Cette dernière valeur est la plus utilisée.
- R pour Response : Cet opérateur correspond au temps de réponse du système. Il est pris en compte seulement s'il entraîne l'attente de l'utilisateur. Il est souvent négligeable dans les applications traditionnelles sur ordinateur de bureau mais moins négligeable sur les dispositifs mobiles [Luo et John, 2005].

On trouve dans la littérature des définitions d'autres opérateurs pour des cas spécifiques. Ils sont souvent introduits pour simplifier des opérateurs de base de KLM comme l'opérateur D (par exemple : pour une entrée de texte gestuelle ou analogique, T_D peut être remplacé par le temps moyen d'accomplir un geste ou de tracer une lettre avec cette méthode).

Pour appliquer le modèle KLM afin d'estimer la vitesse de saisie avec une méthode donnée, on peut procéder de la façon suivante :

- Définir le ou les scénarios possibles pour saisir un échantillon de texte¹,
- Lister les tâches correspondantes (de la saisie de texte uniquement) et définir le meilleur scénario pour accomplir chaque tâche (ou le scénario que l'utilisateur peut naturellement réaliser),
- Déterminer les opérateurs KLM physiques en listant les actions élémentaires de niveau « keystroke » (clics, gestes),
- Inclure l'opérateur cognitif et de réponse du système si nécessaire. Ils apparaissent généralement avant et après la saisie de chaque mot,

¹ Nous appelons un échantillon de texte, une quantité de texte représentant les caractéristiques linguistiques et statistiques d'une langue donnée.

- Appliquer le temps standard de chaque opérateur et additionner les différents temps - opérateurs,
- Diviser la taille de l'échantillon de texte utilisé par le temps résultant pour obtenir le CPS et en déduire le WPM.

2.2. Métriques liées à l'effort

2.2.1. Le nombre de clics/gestes

L'une des métriques les plus rapportées dans la littérature de la saisie de texte est le KSPC (KeyStrokes Per Character). Il définit le nombre moyen d'actions de clic nécessaires pour saisir un caractère avec une méthode donnée.

Comme le WPM, le KSPC peut être déduit expérimentalement ou théoriquement. Dans le premier cas, le flux d'actions effectuées pour saisir un échantillon de texte avec une méthode donnée est enregistré. La longueur de ce flux est divisée par la longueur de l'échantillon de texte selon l'équation ci-dessous pour déduire le KSPC.

$$KSPC = \frac{|Keystrokes|}{|Text|} \quad (1)$$

Pour une prédiction théorique du KSPC, il faut définir le nombre de clics nécessaires pour saisir chaque lettre. Si la méthode évaluée permet de saisir les lettres de différentes manières, on considère pour chaque caractère la façon minimale et la façon maximale pour le saisir (en termes de frappes ou « keystrokes ») pour déduire un KSPC minimal et un KSPC maximal. Le KSPC est ensuite calculé par l'équation suivante qui tient compte des fréquences des lettres dans une langue donnée :

$$KSPC = \frac{\sum(F_c \times K_c)}{\sum F_c} \quad (2)$$

Avec :

- c représente un caractère (notamment une lettre) de l'alphabet utilisé,
- F_c est la fréquence d'un caractère c dans la langue,
- K_c est le nombre de « Keystrokes » nécessaires pour saisir ce caractère.

En utilisant les probabilités d'apparition des lettres dans la langue, l'équation précédente devient :

$$KSPC = \sum(P_c \times K_c) \quad \text{avec :} \quad \sum P_c = 1 \quad (3)$$

On peut aussi calculer le KSPC à partir des fréquences des mots quand le nombre de frappes dépend du mot ou à partir des fréquences de digrammes si la saisie d'un caractère dépend de celui qui le précède [MacKenzie, 2002].

Une autre mesure de performances similaire au KSPC est le GPC (Gesture Per Character). Cette métrique est souvent utilisée dans les cas d'entrées gestuelles comme Graffiti™ ou EdgeWrite™ [Wobbrock et al., 2003]. C'est une extension du KSPC et il peut être calculé de la même manière. Le problème avec cette mesure est la définition même d'un geste. Wobbrock définit un geste comme « toute action atomique effectuée au

cours de la procédure de saisie de texte » et invite les chercheurs à définir clairement ce qu'ils considèrent comme « geste » dans leurs communications [MacKenzie et Tanaka-Ishii, 2007]¹.

2.2.2. Les taux d'erreurs

Les erreurs commises lors de la saisie et leur correction peuvent être considérées comme un effort supplémentaire lors de la saisie. Plusieurs travaux ont été effectués notamment par [Soukoreff et MacKenzie, 2001 ; Soukoreff et MacKenzie, 2004 ; Kano et al., 2007 ; Wobbrock et al., 2008], pour classifier les erreurs de saisie et en définir des mesures. Le KSPC défini dans la section précédente est souvent utilisé comme indicateur sur les erreurs [MacKenzie et Tanaka-Ishii, 2007]² car il prend en considération la totalité des actions effectuées durant la saisie, incluant les actions de suppression et de correction des erreurs en les mettant en rapport avec le texte final produit.

À part le KSPC, il existe des métriques plus spécifiques à la quantification des erreurs. Le taux d'erreurs par MSD (Minimum String Distance) est l'un des plus utilisés. Il permet de quantifier les erreurs présentes dans le texte résultant en le comparant avec le texte à saisir [Soukoreff et MacKenzie, 2001]. Cette méthode se base sur la distance de Levenshtein [Levenshtein, 1966] pour calculer le taux d'erreurs selon l'équation suivante [Soukoreff et MacKenzie, 2001] :

$$TauxD'Erreurs = \frac{MSD(A,B)}{\max(|A|,|B|)} \times 100\% \quad (4)$$

Où :

- A et B sont la chaîne de caractères à saisir et celle effectivement saisie (l'ordre n'est pas important).
- MSD(A,B) est une fonction qui retourne la distance entre les deux chaînes selon l'algorithme repris de [Soukoreff et MacKenzie, 2001] dans la Figure I-3.

L'exemple de la Figure I-3 donne une distance de 2 entre les deux chaînes de caractères. Le taux d'erreurs est donc de 50% dans cet exemple.

Cette mesure du taux d'erreurs basée sur le MSD, contrairement au KSPC, prend en compte seulement la différence entre le texte à saisir et le texte résultant à la fin de la tâche de saisie. MacKenzie et ces collègues ont étudié conjointement le MSD et le KSPC pour redéfinir les taux d'erreurs [Soukoreff et MacKenzie, 2003] en décomposant le flux de la saisie (keystrokes) en plusieurs types de clics :

- C (Correct) : pour les clics qui correspondent aux caractères correctement saisis,
- IF (Incorrect but Fixed) : pour les clics qui correspondent à des erreurs de saisie qui ont été corrigés par la suite,

¹ Chapitre 3, page 53 (par Jacob, O. Wobbrock).

² Chapitre 3, page 52 (par Jacob, O. Wobbrock).

- INF (Incorrect Not Fixed) : pour les clics qui correspondent à la saisie de caractères incorrects,
- F (Fixes) : pour les clics qui correspondent aux corrections, ce qui inclut seulement les actions de suppression et non celles de correction. Ces dernières sont comptabilisées dans C si elles sont définitivement correctes, dans IF si elles sont recorrigées par la suite ou dans INF si elles sont incorrectes et qu'elles le restent.

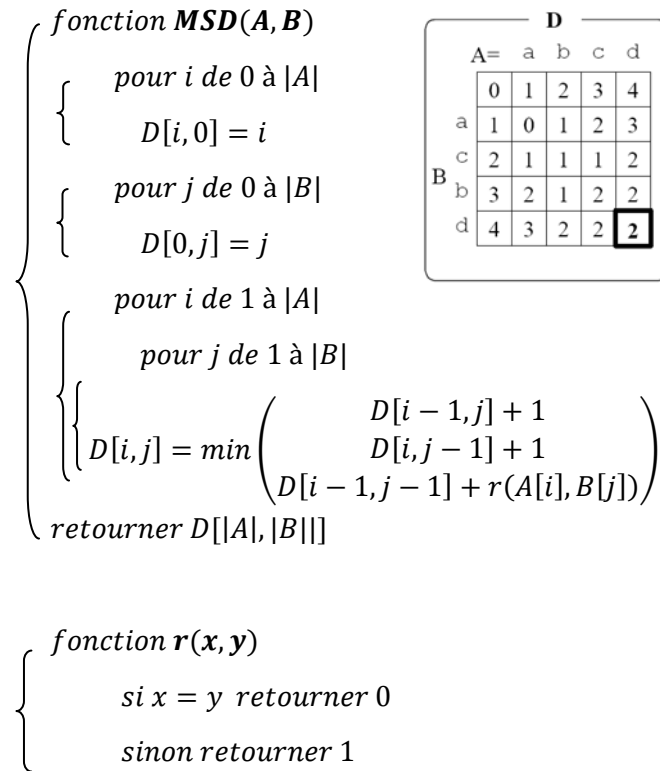


Figure I-3. Algorithme pour calculer la distance de Levenshtein. Exemple avec les deux chaînes de caractères « abcd » et « acbd ».

Le taux d'erreurs MSD et le KSPC se présentent donc sous la forme suivante :

$$MSD_{ER} = \frac{INF}{C + INF} \times 100\% \quad (5)$$

$$KSPC = \frac{C + INF + IF + F}{C + INF} \quad (6)$$

D'autres données statistiques ont été définies par MacKenzie et ces collègues en se basant sur cette décomposition. On en note essentiellement les suivantes :

- CE (Correction Efficiency) : pour l'efficacité de la correction : $CE = \frac{IF}{F}$
- PC (Participant Conscientiousness) : pour le ratio des erreurs corrigées sur la totalité des erreurs commises : $PC = \frac{IF}{INF + IF}$
- NCER (Not Corrected Error Rate) : pour le taux d'erreurs non corrigées :

$$NCER = \frac{INF}{C+INF+IF} \times 100\%$$

- CER (Corrected Error Rate) : pour le taux d'erreurs corrigées :

$$CER = \frac{IF}{C+INF+IF} \times 100\%$$

- TER (Total Error Rate) : pour le taux d'erreurs total : $TER = \frac{INF+IF}{C+INF+IF} \times 100\%$

Ces métriques ont été basées sur des études de méthodes de saisie à un clic par caractère. Une extension de ces métriques d'erreurs a été proposée ensuite pour les adapter aux méthodes « multitap »¹ et pour en introduire d'autres comme le « Corrected but Right Error Rate » et le « Corrected and Wrong Error Rate » respectivement pour les corrections effectuées sur des caractères correctement saisis à la base et les corrections effectuées sur des caractères erronés [Soukoreff et MacKenzie, 2004].

Les métriques d'erreurs proposées dans les différents travaux de MacKenzie et Soukoreff constituent une référence pour la recherche dans le domaine mais elles ne couvrent pas tous les types de méthodes de saisie de texte car elles se basent sur une analyse de niveau clic (keystroke) ou parfois de niveau caractère et elles ne considèrent pas d'autres types de clics comme ceux qui correspondent à des fonctions ou commandes diverses [Gong et Tarasewich, 2006] (par exemple : le clic sur la touche « suivant » dans une méthode prédictive).

2.3. Autre métriques

Les métriques citées dans les sections précédentes ne permettent pas d'évaluer toutes les performances relatives à la saisie de texte. Les chercheurs essayent toujours de trouver des moyens pour cerner quantitativement plus de caractéristiques d'une méthode de saisie de texte. L'une des recherches récentes dans notre équipe au laboratoire Valoria dans le cadre de la thèse de Hamed Sad consiste à quantifier la charge mentale que suscite une méthode donnée pour saisir du texte [Poirier et Sad, 2007]. Une mesure de performance (PM) a été ainsi définie à partir de mesures classiques comme la vitesse empirique de saisie et le taux d'erreurs obtenus lors d'une saisie avec et sans « tâche secondaire ». Cette mesure permet donc d'évaluer l'influence de la tâche secondaire sur les performances de la méthode de saisie étudiée. La tâche secondaire est définie dans les expériences menées par Sad et Poirier en tant que tâche auditive qui nécessite une attention variable de la part de l'utilisateur et réglable par l'évaluateur pour pouvoir tester les méthodes de saisie avec différents niveaux d'importance de la tâche secondaire. Une plateforme d'évaluation des méthodes de saisie pour dispositifs mobiles développée aussi dans le cadre du même projet intègre cette nouvelle mesure de performance. Cette plateforme permet d'évaluer diverses méthodes et d'avoir des résultats subjectivement comparables [Poirier et Sad, 2008].

¹ Méthodes de saisie qui consistent à effectuer plusieurs appuis successifs sur la même touche pour saisir un caractère (comme le clavier téléphonique classique).

En plus des métriques citées précédemment, il existe d'autres métriques moins utilisées ou utilisées dans des contextes plus spécifiques. Nous en listons quelques unes dans le tableau suivant :

Métrique	Signification	Références
GPS	Gesture Per Second	[MacKenzie et Tanaka-Ishii, 2007] ¹
GPT	Gesture Production Time	[Cao et Zhai, 2007]
KSPS	Key Stroke Per Second Souvent appelé aussi : Key Repeat Rate ou Key Press Rate	[MacKenzie et Tanaka-Ishii, 2007] ² [Wobbrock, 2006] [Pavlovych et Stuerzlinger, 2005]
CPW	Clics Per Word	[Luoma, 2003]
KSPW	Key Stroke Per Word	[MacKenzie et Tanaka-Ishii, 2007] ³
CWPM	Correct Word Per Minute	[Goldstein et al., 1999]
CIR	Corrected Insertion Rate	[Wobbrock et Myers, 2006a]
CPC	Cost Per Correction	[Gong et Tarasewich, 2006]
KSR	Keystroke Saving Rate	[Wandmacher et al., 2008]
MPM	Mots Par Minute (traduction de WPM)	[Martin, 2005]

Tableau I-1. Autres métriques utilisées pour l'évaluation des méthodes de saisie de texte.

Les références citées dans ce tableau ne présentent pas forcément les métriques concernées mais les utilisent pour évaluer une méthode donnée.

3. État de l'art des méthodes de saisie en mobilité

Nous appelons « méthode de saisie de texte en mobilité » tout système ou dispositif permettant d'entrer du texte ou des données en général sur un appareil ou un dispositif électronique destiné à être utilisé dans des conditions nomades (sans file, mobile). Vu le grand nombre de méthodes de saisie qui existent (ou qui ont existé), nous présentons dans la suite, les méthodes de saisie les plus récentes (publiées ou commercialisées pendant la durée de cette thèse) ainsi que les plus référencées.

La classification des méthodes de saisie la plus basique consiste à les diviser en deux catégories : les méthodes de saisie à touches et les méthodes gestuelles [Martin et Pecci, 2007] ou bien les méthodes de saisie à touches et les méthodes de saisie à stylet [MacKenzie et Soukoreff, 2002]. Dans la suite, pour décrire quelques méthodes de saisie, nous nous inspirons de ces classifications en divisant ces méthodes en trois catégories selon la modalité d'interaction : les méthodes à touches physiques, les méthodes à interface tactile et les méthodes à détection du mouvement.

¹ Chapitre 3, page 51 (par Jacob O. Wobbrock).

² Chapitre 3, page 50 (par Jacob O. Wobbrock).

³ Chapitre 5, page 110.

3.1. Méthodes à touches physiques

3.1.1. Le clavier téléphonique T9

Développée par Tegic Communications en 1998 (actuellement Nuance Communications¹), la principale caractéristique de la méthode de saisie T9 est qu'elle est la première à remplacer la saisie classique « Multitap », permettant aux utilisateurs de composer leurs textes avec un seul appui de touche par caractère (1 KSPC) sur leurs téléphones mobiles [Grover et al., 1998]. Chaque touche du clavier téléphonique standard (connu aussi sous le nom T12) est libellée au minimum par 3 lettres. La saisie multitap permet d'entrer, par exemple, la lettre 'a' en un appui sur la touche '2', la lettre 'b' en deux appuis et la lettre 'c' en trois appuis toujours sur la même touche. T9 fait de sorte qu'un simple appui sur la touche '2' par exemple permette de saisir soit la lettre 'a', 'b' ou 'c'. À chaque appui, les mots correspondants à la séquence de touches appuyées sont automatiquement inférés d'une base de mots de la langue utilisée (dictionnaire). L'utilisateur peut ensuite, par exemple, saisir un espace pour valider le premier mot affiché dans la liste ou défiler cette liste en utilisant les touches de navigation (pavé directionnel) ou la molette de l'appareil afin de choisir le bon mot. La saisie avec cette méthode est dite « ambiguë » car un appui sur une touche désigne une lettre parmi plusieurs sans la valider. La validation est faite au niveau mot.

L'un des problèmes qui ont freiné la méthode T9 est que l'utilisateur se trouve bloqué quand il veut saisir un mot qui n'existe pas dans le dictionnaire utilisé. Dans ce cas, l'utilisateur doit effectuer une séquence de suppressions pour effacer le mot saisi, puis entrer le nouveau mot dès le début en utilisant une autre technique, en l'occurrence Multitap. Kober et al., mentionnent que si on évite les hypothèses communément utilisés pour le T9 (par exemple, la non considération des erreurs de saisie), les performances du T9 baissent de manière significative [Kober et al., 2001]. Ce problème a été corrigé plus tard en permettant de basculer facilement en mode multitap sans effectuer des suppressions pour ajouter un mot au dictionnaire et le proposer par la suite dans les résultats de désambiguïsation.

T9 est l'une des méthodes non académiques² les plus référencées dans la littérature de la saisie de texte. Elle a été étudiée dans plusieurs contextes en comparaison avec d'autres méthodes, essentiellement Multitap et LetterWise [MacKenzie et al., 2001]. Voici quelques performances (niveau expert) de la méthode de saisie T9 rapportées dans les travaux de recherche :

- KSPC Théorique :
 - 1,0072 [MacKenzie, et al., 2001]
- KSPC expérimental :
 - 1,0088 (texte de journal) [Hasselgren et al., 2003]

¹ Nuance Communications, page Web de T9 : www.nuance.com/t9/textinput/, visitée le 29/01/2009.

² Nous entendons par « non académique » une méthode qui a été brevetée et commercialisée par un organisme privé sans ou avant d'être publiée dans une thèse, dans des actes ou dans une revue du domaine.

- 1,0806 (SMS) [Hasselgren et al., 2003]
- WPM Théorique :
 - 40,6 – 45,7 [James et Reischel, 2001]
- WPM expérimental :
 - 21,7 [MacKenzie et al., 2001]
 - 15,05 (texte de journal) [James et Reischel, 2001]
 - 25,68 (discussion instantanée) [James et Reischel, 2001]

Avec le développement des PDA et des téléphones intelligents, Nuance Communications a développé aussi une autre version de T9 appelée XT9 qui s'intègre aux méthodes de saisie utilisées par le dispositif (reconnaissance des lettres, clavier logiciel AZERTY et clavier T12 par exemple). Quand un clavier ambigu est utilisé, XT9 assure la tâche de désambiguïsation comme le T9 classique. De plus, XT9 fait de la complétion de mots en prédisant la fin du mot à partir des premières lettres saisies.

On note aussi d'autres méthodes similaires à XT9 qui combine la désambiguïsation avec la complétion des mots comme eZiText™ et eZiType™ de Zi Corporation¹ et iTap™ de Motorola². eZiText fait, en plus de la désambiguïsation, de la prédiction pour compléter le mot en cours de saisie ou même le compléter et prédire le mot qui suit. Elle s'adapte aussi au vocabulaire de l'utilisateur. eZiText est présentée comme un outil efficace pour saisir des SMS. eZiType est plus adaptée à la saisie des mails. En plus de la désambiguïsation et la complétion, elle corrige automatiquement les erreurs de saisie ou d'orthographe communes. La particularité d'iTap réside dans l'association du vocabulaire de l'utilisateur avec l'application utilisée. Lors de la saisie du texte, le système de prédiction met en avant les mots qui ont été saisis précédemment sur la même application (par exemple : fenêtre SMS, client mail ...). iTap comme eZiText, eZiType et XT9 se présentent sous forme de module logiciel utilisable avec toute méthode de saisie ambiguë implémentée sur le dispositif.

3.1.2. Le clavier Q12

Breveté en 2006 par SOFTAVA³, le Q12 est un clavier téléphonique basé sur une configuration traditionnelle à 12 touches : 10 touches numériques, une pour '#' et une pour '*'. Les lettres sont imprimées sur la carcasse du téléphone entre les touches. Deux distributions de lettres sont proposées : une organisation alphabétique appelée « Classic » et une organisation QWERTY appelée « Computer ». Pour saisir une lettre, l'utilisateur doit appuyer simultanément (au doigt) sur les deux ou les trois touches qui entourent cette lettre (voir Figure I-5). Par exemple pour saisir la lettre 'a' avec le clavier Q12 Computer, il faut combiner les touches 4 et 7 et pour saisir la lettre 's', il faut combiner les touches 4, 5 et 7.

¹ Zi Corporation, page Web des méthodes de saisie : www.zicorp.com/TextEntry.htm, visitée le 29/01/2009.

² Motorola Inc., Division Lexicus, site web : www.motorola.com, visité le 29/01/2009.

³ Softava Inc., site web : www.softava.com, visité le 29/01/2009.

Le précurseur du clavier Q12 s'appelle Fastap [Levy, 2002]. Il est composé de petites touches alphabétiques en plus des touches numériques. Puisque les touches sont de petites dimensions (environ 3,3 touches par cm² en moyenne contre 1,2 touche par cm² dans les claviers T12), une technique appelée « passive chording » est utilisée pour éviter les erreurs en cas d'appui simultané sur les touches qui entourent celle désirée. Q12 a repris ce principe en passant à un accord actif (active chording).



Figure I-4. Le clavier Q12 Classic avec son organisation alphabétique et le clavier Q12 Computer avec son organisation QWERTY.

L'avantage principal apparent de ce clavier est que l'appui sur les touches est simultané. Il présente aussi l'avantage d'être basé sur un clavier familier aux utilisateurs de téléphones mobiles et d'utiliser des dispositions de caractères connues. Aucune étude de performances n'a été publiée sur ce clavier à ce jour. Le principal désavantage pourrait être un important taux d'erreurs de saisie car, malgré les formes adaptées des touches, l'appui simultané sur plusieurs touches se fait en un seul doigt, ce qui peut engendrer des erreurs dans les cas où l'une des touches désignées n'est pas suffisamment appuyée.

3.1.3. CyKey

CyKey est un clavier à accord (chord) commercialisé par Bellaire Electronics¹ en 2006. Il permet de saisir du texte sur PocketPC, Palm ou PC à condition de disposer d'une borne infrarouge. Le principe de l'accord dans la saisie de texte consiste à appuyer sur plusieurs touches simultanément pour produire un caractère. Ce principe n'est pas nouveau. Il est souvent illustré par Twiddler, une méthode de saisie très populaire dans la communauté de l'informatique vestimentaire [Lyons et al., 2004]. Dans le cas de CyKey, la saisie se fait en une seule main en s'appuyant sur la carte des caractères de son ancêtre « Microwriter ». Ce dernier est un petit appareil qui a été commercialisé au début des années 80 pour la prise de notes. Microwriter a essayé de remplacer le clavier QWERTY encombrant et il a donné naissance à l'une des premières méthodes de saisie pour dispositifs mobiles. L'alphabet utilisé se base sur des petites phrases pour chaque lettre pour faciliter l'apprentissage (voir Figure I-6). Par exemple pour la lettre 'F' la

¹ Bellaire Electronics, site web : www.bellaire.demon.co.uk/, visité le 29/01/2009.

phrase caractéristique est « First Four Fingers » pour dire que cette lettre correspond aux 4 touches de la gauche. Ces phrases métaphoriques ne sont pas toujours précises.



Microwriter, un dispositif électronique mobile pour la prise de notes (1980)



CyKey, un clavier à accord pour dispositifs mobiles inspiré de Microwriter.



L'alphabet Microwriter.

Figure I-5. Le clavier CyKey, son ancêtre et sa table de caractères.

Les principales qualités de CyKey sont le nombre réduit de touches et les grandes dimensions de ces dernières. Cependant, il reste un clavier externe encombrant et il ne peut être utilisé qu'en situation d'immobilité sur un bureau par exemple. À notre connaissance, aucune étude scientifique n'a été faite pour évaluer ce clavier.

3.1.4. EQx

Le clavier EQx (Eatoni¹ QWERTY with x columns) est une famille de claviers téléphoniques ambigus (comme le clavier T9) qui utilisent une disposition QWERTY (comme SureType² par exemple). La différence réside dans l'adaptabilité au nombre de touches disponibles sur le clavier du dispositif. L'agencement des lettres sur les touches est fait de façon à garder le plus fidèlement possible la disposition QWERTY tout en optimisant le résultat de la désambiguïsation. La Figure I-7 montre le clavier EQ3

¹ EATONI Ergonomics, page web de EQx : <http://www.eatoni.com/wiki/index.php/EQx>, visitée le 29/01/2009.

² BlackBerry®, page web de SureType : <http://www.blackberry.com/la/products/suretype/index.shtml>, visitée le 29/01/2009.

intégré au Nokia 6682 et le clavier EQ6 présenté en démonstration sur le site web de Eaton.

Les claviers EQx se distinguent aussi des claviers ambigus classiques au niveau de la saisie des mots qui n'existent pas dans le dictionnaire utilisé. La méthode LetterWise [MacKenzie et al., 2001] est utilisée dans ces cas au lieu du Multitap. Cette méthode change dynamiquement l'ordre des lettres sur chaque touche ambiguë en fonction du préfixe saisi.



Figure I-6. Exemples de claviers EQx.

Une prédiction théorique des performances de ces claviers publiée par Castellucci [2007] dans son mémoire de Master à l'université de York donne les résultats suivants :

- EQ3 :
 - KSPC : 1,0023
 - WPM : 46,7 (British National Corpus)
 - WPM : 43,6 (SMS)
- EQ6 :
 - KSPC : 1,0001
 - WPM : 46,0 (British National Corpus)
 - WPM : 43,2 (SMS)

3.2. Méthodes à interface tactile

3.2.1. Les claviers logiciels complets

On entend par claviers logiciels ou virtuels complets, les claviers qui affichent au moins toutes les lettres de l'alphabet attribuées chacune à une touche et saisissables en un simple clic. Le plus connu de ces claviers est sans doute celui hérité de l'ordinateur de bureau avec la traditionnelle organisation AZERTY (ou QWERTY). Cette organisation des touches a été créée à l'origine pour la dactylographie en 1874, elle a survécu et elle est encore utilisée à nos jours.

Quand on plonge dans l'histoire, on découvre que les premières machines à écrire utilisaient une organisation alphabétique avant que Christopher Sholes et Amos Densmore ne se rendent compte qu'elle cause des conflits à cause des croisements des barres à caractères lorsque la saisie est rapide. Ils décidèrent alors d'éloigner les lettres qui se succèdent entre elles en se basant sur des données linguistiques recueillies au sein de leur entreprise. L'organisation QWERTY a été donc énormément influencée par les

contraintes mécaniques des liaisons des barres à caractères à l'intérieur de la machine. Le nouveau clavier a démontré une plus grande vitesse de saisie et les dactylographes en ont été satisfaits. Il a été donc breveté ¹ en 1878, quelques années après sa commercialisation sur les machines à écrire. Le clavier QWERTY s'est adapté ensuite aux différentes langues puis, avec l'arrivée de l'informatique, à l'ordinateur qui a fait de ce clavier son périphérique d'entrée principal.

Actuellement, avec le développement de l'informatique mobile, les règles du jeu ont changé. L'espace est de plus en plus rare et les contraintes de la mobilité et des nouveaux modes d'interaction imposent souvent qu'une seule main soit utilisée pour la saisie. Cependant, les claviers du type QWERTY, grâce à leur familiarité, sont encore implémentés sur beaucoup de dispositifs mobiles dès que les dimensions le permettent. On cite notamment le clavier virtuel de Microsoft Windows Mobile® livré sur un grand nombre de Pocket PC. Les dimensions des touches ont été réduites au minimum et les appuis sur les touches sont effectués à l'aide d'un stylet. Selon la loi de Fitts, plus la distance est grande et la largeur de la cible est réduite, moins l'interaction est rapide. MacKenzie et al., [1999] rapportent une vitesse de 22,9 WPM avec un clavier virtuel QWERTY mais plus la vitesse de saisie augmente plus les erreurs sont fréquentes. Plusieurs travaux ont été menés pour optimiser le clavier virtuel par rapport à la loi de Fitts en se basant sur des données statistiques du langage, le moyen étant dans la plupart des cas un agencement des touches de manière à rapprocher celles correspondantes aux caractères qui se succèdent le plus fréquemment. Le Tableau I-2 montre les performances connues des principaux claviers logiciels optimisés par agencement.

En plus de la réorganisation des lettres sur la surface du clavier, d'autres techniques sont utilisées pour mieux adapter les claviers complets à une utilisation nomade. Parmi ces techniques :

- L'utilisation de feedbacks visuels dynamiques, appelés aussi « indices visuels » par Magnien et al., [2003] : les lettres les plus probables à saisir après un préfixe donné sont mises visuellement en valeur pour optimiser le temps de la recherche visuelle.
- L'utilisation des « pie-menus » pour afficher à chaque appui de touche, les lettres les plus probables à saisir, autour de la lettre appuyée. Cela permet de réduire le temps de la recherche visuelle et le temps de mouvement pour saisir la lettre suivante. Un exemple récent de l'utilisation de cette technique est le clavier KeyGlass [Raynal, 2007]. Il affiche quatre touches semi-transparentes, contenant les lettres les plus probables à saisir, autour de la touche appuyée.
- L'exploitation du geste : avec les interfaces tactiles, une nouvelle possibilité s'ouvre devant les claviers logiciels. Les gestes utilisés peuvent être divisés en deux catégories : une première qui est très proche du principe des pie-menus où chaque direction gestuelle correspond à un caractère supplémentaire (en plus du

¹ Brevet US numéro 207559, datant d'août 1878, par Latham SHOLES.

caractère principal de la touche) comme dans le cas de la méthode 2CPS (Two Characters Per Stroke) [Jhaveri, 2003]. La deuxième catégorie concerne les gestes analogiques comme dans le cas du clavier logiciel de Windows Mobile® où un geste vers la droite correspond à un espacement, un geste vers la gauche correspond à une suppression, un geste vers le bas correspond à un retour à la ligne et un geste vers le haut permet de saisir la lettre correspondante à la touche sélectionnée en majuscule.

- L'utilisation du zoom : vu que les dimensions des touches dans un clavier logiciel complet pour dispositifs mobiles sont très réduites, la saisie devient difficile et les erreurs de saisie surgissent en masse. L'utilisation d'un zoom sur le point de contact du stylet avec l'interface permet de bien voir la touche appuyée. La saisie de la lettre correspondante est validée au lever du stylet. Un exemple récent de l'utilisation de cette technique est le clavier de l'iPhone [Sadun, 2008] et son homologue PCM Keyboard¹ pour Windows Mobile®. Le clavier FishEye [Raynal et Truillet, 2006] par exemple effectue un agrandissement de la touche sélectionnée et de son entourage. Cette technique peut réduire les taux d'erreurs mais au prix de la vitesse de la saisie puisque le temps de la réalisation du zoom et le temps de la réaction de l'utilisateur à ce zoom sont ajoutés.
- Améliorations « esthétiques » : en partant du principe que l'esthétique joue un rôle déterminant dans la perception des systèmes et l'efficacité des interactions, on suggère que l'esthétique est un fort déterminant du plaisir qu'éprouve l'utilisateur au cours de l'interaction. Une étude sur l'impact des modifications « purement » esthétiques (voir [Wong et al., 2007]) sur les performances de la saisie avec un clavier logiciel complet montre que les performances chutent considérablement avec ces nouveaux designs.

Malgré les différentes améliorations apportées aux claviers logiciels complets, ils restent encombrants et ne peuvent être implémentés que sur les PDA ou les dispositifs mobiles de plus grandes dimensions.

¹ www.pocketcm.com, visité le 29/01/2009.

Clavier	Caractéristiques	Performances connues	Références	Remarques
OPTI	- Forme rectangulaire, - Quatre touches du caractère « espace ».	WPM théorique : 58,2 WPM empirique : 44,3 Taux d'erreurs : 4,18%	[MacKenzie et Zhang, 1999]	- Le taux d'erreurs augmente avec la vitesse de saisie. - Les prédictions et expériences sont réalisées sur un Tablet PC.
		WPM théorique : 40,3	[Zhai et al., 2002a]	
FITALY	- Solution commerciale, par Textware Solutions. - Deux touches d'espacement (chacune fait la taille de 2 touches normales)	WPM empirique : 8,2 (Novice)	[MacKenzie et al., 1999]	- Les valeurs données par Textware Solutions montrent une vitesse expérimentale maximale de 81,7 WPM, - Des résultats antérieurs donnés par la firme ont été mis en doute [Zhai et al., 2002a]. - L'étude de MacKenzie et al., qui a donné 8,2 WPM pour FITALY affiche 20,2 pour le clavier QWERTY.
		WPM empirique : 57,7	Site web de FITALY ¹	
		WPM théorique : 35,2 – 37,07	[Zhai et al., 2002a]	
Metropolis	- Cet agencement est produit à l'aide d'un algorithme de « marche aléatoire » qui permet de rapprocher progressivement les lettres fréquemment voisines dans une langue donnée,	WPM théorique : 43,1	[Zhai et al., 2000]	- Les prédictions des performances donnée par Zhai et ses collègues est faite pour une implémentation de Metropolis sur Tablet PC.

¹ Textware Solutions, page web décrivant un concours de saisie de texte avec FITALY, Graffiti, Jot et le clavier QWERTY : www.fitaly.com/domperignon/domperignon2.htm, visitée le 29/01/2009.

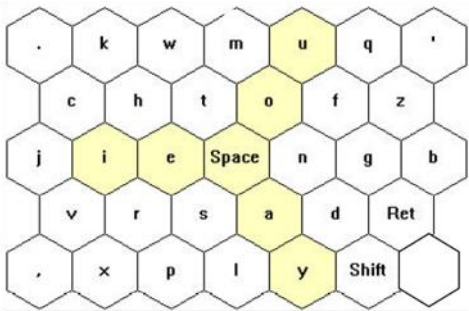
	- Toutes les touches sont disposées autour de celle de l'espace (caractère le plus fréquent), - un clavier peut être facilement généré pour une langue donnée. Les seuls pré-requis sont la liste des caractères et les fréquences des bigrammes, - Touches hexagonales.	WPM théorique : 41,6 – 43,1	[Zhai et al., 2002a]	
ATOMIK	- Alphabetically Tuned and Optimized Mobile Interface Keyboard, - Généré de la même façon que Metropolis, - En plus de l'optimisation de l'efficacité du mouvement, des optimisations alphabétique et de connectivité des mots les plus fréquents, - Breveté par IBM¹.	WPM théorique : 39,9	[Zhai et al., 2002a]	- Prédiction pour implémentation sur Tablet PC.
		WPM théorique : 45,3	[Zhai et al., 2002b]	
GAG	- Généré par Algorithme Génétique.	WPM théorique : 45,5 – 46,4	[Raynal, 2006]	

Tableau I-2. Exemples de claviers logiciels optimisés par agencement.

¹ Brevet US numéro 7251367 datant de Décembre 2002, par Shumin Zhai.



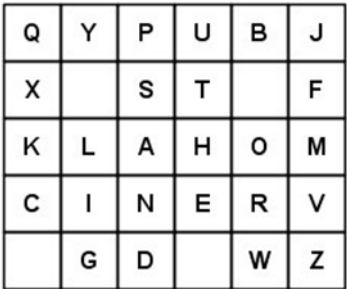
Clavier OPTI



Clavier Metropolis



Clavier FITALY



Claviers GAG



Clavier ATOMIK

Figure I-7. Exemples de claviers logiciels optimisés par agencement.

3.2.2. Écriture manuscrite

L'écriture manuscrite est le mode le plus « naturel » pour la saisie de texte, grâce à l'expérience acquise par les utilisateurs avec l'écriture sur papier. La technologie de la reconnaissance d'écriture a fait des progrès remarquables ces dernières années mais les taux d'erreurs restent élevés. À l'heure actuelle, la tendance est d'utiliser des alphabets de caractères plus au moins simplifiés comme Graffiti et Jot pour une meilleure reconnaissance [Goldberg et Richardson, 1993] mais cela n'est pas suffisant pour réduire les taux d'erreurs. Fleetwood et al., [2002] montrent que les taux d'erreurs ne baissent pas avec la pratique et qu'ils restent largement supérieurs à celui observé avec un clavier virtuel.

En plus des erreurs, la faiblesse fondamentale de l'écriture manuscrite reste la limitation de la vitesse de saisie, généralement estimée à près de 15 WPM [Card et al., 1983]. Pour Graffiti et Jot, Sears et Arora [2002] ont constaté entre 4,3 et 7,7 WPM avec des utilisateurs novices et entre 14 et 18 WPM avec des utilisateurs expérimentés, bien que d'autres rapports informels annoncent des performances supérieures. Ces vitesses pourraient être suffisantes pour saisir des noms et des numéros de téléphone sur un PDA, mais restent lentes pour la rédaction d'un texte plus long. FreePad reste une méthode de saisie par reconnaissance d'écriture alors que beaucoup de chercheurs du domaine comme Poika Isokoski [2004]¹ mettent en doute l'efficacité de cette technique pour la saisie de texte sur dispositifs mobiles même si on arrive à trouver un algorithme qui permet une saisie à « presque zéro erreurs ».

3.2.3. ShapeWriter

Développée au début par des chercheurs d'IBM Almaden Research Center sous le nom de « SHARK » (pour SHorthand Aided Rapid Keyboarding) [Zhai et Kristensson, 2003 ; Kristensson et Zhai, 2004], ShapeWriter est une méthode de saisie sténographique qui permet à l'utilisateur d'écrire du texte sous forme de « sokgraphs », une forme de sténographie définie sur un clavier logiciel en reliant les différentes lettres du mot à saisir. Le clavier utilisé au départ est ATOMIK Keyboard, un clavier logiciel optimisé au niveau de la distribution des caractères et de la forme des touches [Zhai et al., 2002a]. ShapeWriter a été déclarée ensuite comme *une méthode indépendante du calque* [Zhai et Kristensson, 2006]. Elle peut être utilisée avec le clavier ATOMIK comme avec le clavier QWERTY ou toute autre disposition optimisée ou non. Cela implique bien sûr que la carte des « sokgraphs » annoncés pour SHARK ne soit plus la seule disponible mais elle reste la plus optimale. Le clavier ATOMIK a aussi été retouché pour mieux l'adapter à ShapeWriter [Zhai et Kristensson, 2006]. Le clavier QWERTY a aussi été adapté récemment sous le nom de iQWERTY (interlaced QWERTY) et a donné de meilleurs résultats en terme de la différenciation des sokgraphs et du temps de recherche visuelle par rapport aux autres organisations de claviers [Zhai et Kristensson, 2008].

L'avantage principal de ShapeWriter est que la saisie est rapide vu que l'utilisateur n'est pas obligé de relier strictement les lettres qui composent le mot à saisir mais plutôt

¹ Section « Text Recognition ».

de dessiner le « sokgraph » correspondant sur l'interface. ShapeWriter utilise un mécanisme de reconnaissance des formes pour déduire les mots dessinés. Le problème majeur de cette méthode est que l'apprentissage même de ces formes « sokgraphs » par l'utilisateur n'est pas facile. En effet, d'autres méthodes de saisie gestuelles ou « unistroke » sont souvent critiquées pour ce problème même si elles proposent un feedback visuel. Le clavier Glyph2PPC (voir chapitre II) développé au VALORIA ou la méthode Unistrokes de Xerox [Goldberg et Richardson, 1993] par exemple sollicitent l'utilisateur à mémoriser au maximum une trentaine de gestes correspondants aux lettres tout en lui proposant un principe d'analogie et/ou un feedback visuel. Dans le cas du « Shape Writing » l'utilisateur doit mémoriser un geste plus ou moins complexe pour chaque mot en plus des gestes correspondants aux différentes commandes de l'édition de texte [Kristensson et Zhai, 2007]. Cela dépasse très largement la trentaine de gestes à mémoriser dans le cas des autres méthodes tant en nombre qu'en complexité de la forme. Les auteurs justifient la possibilité de mémoriser autant de symboles par le fait que *700 hiéroglyphes distincts étaient utilisé dans l'ancienne Égypte pour écrire et qu'un chinois lettré doit apprendre deux à cinq milliers de caractères uniques* [Zhai et Kristensson, 2003].

Concernant les performances, les auteurs démontrent dans plusieurs publications l'efficacité de cette méthode sans utiliser les métriques conventionnelles. Dans [Zhai et Kristensson, 2003] par exemple, le WPM est utilisé dans l'état de l'art mais les auteurs n'y reviennent plus dans la présentation et l'évaluation de la méthode. Une étude comparative de ShapeWriter avec Graffiti et Unistrokes en termes de « Gesture Production Time » [Cao et Zhai, 2007] donne les résultats décrits dans le Tableau I-3.

	Unistrokes	ShapeWriter (ATOMIK)	ShapeWriter (QWERTY)	Graffiti
Modèle prédictif :	622 ms	600 ms	768 ms	1125 ms
Résultat expérimental :	365 ms	438 ms	506 ms	591 ms

Tableau I-3. Comparaison des méthodes ShapeWriter, Unistrokes et Graffiti en termes de « Gesture Production Time ».

3.2.4. Claviature

Développé par Microth¹ en 2004, Claviature est une méthode de saisie multilingue pour dispositifs mobiles à écran tactile. Douze touches larges sont disposées en deux lignes. À chaque touche sont attribuées 4 lettres et une fonction (retour à la ligne, espacement, suppression, tabulation, etc.) avec une disposition similaire à celle du clavier d'ordinateur de bureau dans la langue utilisée (AZERTY pour le français, QWERTY pour l'anglais ou ЙЦУКЕХ pour le russe par exemple).

Pour saisir un caractère, l'utilisateur doit poser le doigt sur la touche qui contient ce caractère, puis effectuer un geste court correspondant à son emplacement sur la touche. La Figure I-9 montre l'exemple de la saisie de la lettre 'a' : un appui sur la première touche en haut à gauche et un geste vers le bas-gauche.

¹ Microth Inc., site web : www.microth.com, visité le 29/01/2009.

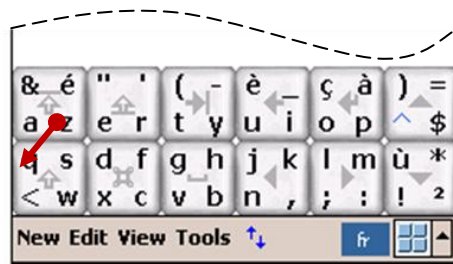


Figure I-8. L'interface de Claviature. Exemple de la saisie de la lettre 'a'.

Ce clavier présente deux grands avantages : le premier est l'utilisation des touches larges avec autant d'espace sur l'écran que le clavier normale miniaturisé, le deuxième est l'utilisation d'une disposition familière à l'utilisateur pour minimiser l'apprentissage. Le nombre de touches reste cependant élevé et le clavier occupe un quart de l'écran du PDA ce qui ne permet pas son implémentation sur des dispositifs de plus petites tailles. Claviature rejoint aussi le clavier normal miniaturisé au niveau de l'encombrement de l'interface puisqu'il affiche 60 symboles (12 × 5).

3.2.5. PNLH Multitap

PNLH (Predictive Next-Letter Highlighting) est un clavier logiciel multitap qui utilise l'organisation classique des lettres du clavier téléphonique T12. Il est très proche de LetterWise [MacKenzie et al., 2001] par son principe prédictif avec l'utilisation d'un modèle de fréquences des trigrammes. L'apport principal est un feedback visuel. Les lettres les plus probables à saisir après un préfixe donné sont mises visuellement en valeur ainsi que les touches qui les contiennent sont mises en surbrillance comme le montre la Figure I-10. Quand l'utilisateur clique sur l'une des touches mises en surbrillance, c'est la lettre en gras qui s'affiche en premier. Des clics successifs permettent d'afficher les autres lettres à la manière du multitap. La touche « Done » est utilisée dans le prototype d'évaluation pour valider le texte saisi et vider la fenêtre d'édition. La touche « Next » est utilisée pour arrêter la temporisation et valider la lettre affichée comme dans le clavier multitap classique.



Figure I-9. L'interface du prototype du clavier multitap PNLH.

Les études menées par les auteurs de la méthode dans [Gong et al., 2005] affichent les performances suivantes :

- KSPC : 1,47 (théorique)
- WPM: 9,16 (empirique)
- Taux d'erreurs : 3,39% (empirique)

3.2.6. VirHKey

VirHKey (VIRtual Hyperbolic KEYboard) est un clavier gestuel de type unistroke. La version publiée permet de saisir les caractères sur un pavé tactile avec un feedback visuel optionnel sur l'écran du dispositif. Chaque caractère correspond à un geste simple ou un geste continu composé d'une succession de gestes simples comme le montre la Figure I-11.

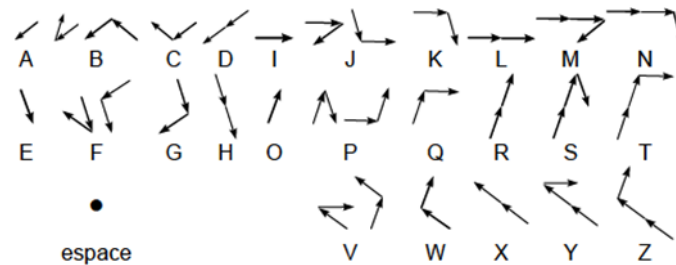


Figure I-10. L'alphabet VirHKey. Le caractère espace correspond à un simple clic, les voyelles correspondent à un geste simple et les autres lettres correspondent à des gestes composés. Certaines lettres sont accessibles par différents chemins.

Le feedback visuel est de type focus et contexte [Martin, 2005] basé sur une géométrie hyperbolique (voir Figure I-12). Quand l'utilisateur pose le stylet sur la surface d'interaction, le point de contact est traduit par une position neutre sur le feedback. Cette position centrale correspond à la saisie d'un espace. Si l'utilisateur n'effectue donc aucune translation avec le stylet, le caractère « espace » est entré au lever du stylet. Sinon, chaque translation (geste simple) permet d'accéder à une lettre de niveau plus profond dans l'arbre hyperbolique. Quand une nouvelle lettre est atteinte, le pentagone (touche pentagonale) correspondant est centré dans l'écran avec un effet de zoom pour donner accès aux lettres suivantes. Les lettres sont organisées dans une hiérarchie hyperbolique par groupes {ABCD, EFGH, IJKLMN, OPQRST, UVWXYZ}. Les voyelles (les 5 voyelles qui délimitent les groupes) sont placées autour du caractère « espace » au premier niveau après la racine et permettent chacune l'accès aux lettres de son groupe.

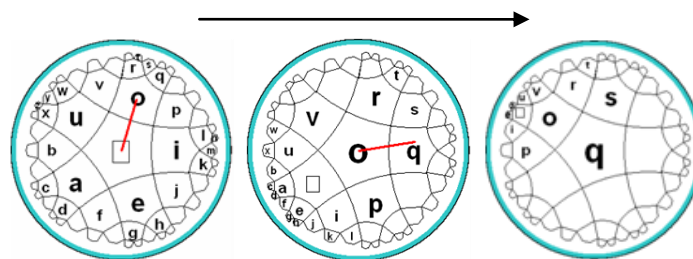


Figure I-11. Exemple de la saisie de la lettre 'q' avec la méthode VirHKey.

Une évaluation expérimentale menée avec 4 sujets montre qu'après 7 heures d'entraînement, les utilisateurs n'ont pas atteint un niveau « expert » (la vitesse de saisie n'a pas convergé) et qu'ils arrivent à saisir 21,78 mots par minute (WPM) avec un taux d'erreur moyen de 5,18% [Martin, 2005]. Le feedback visuel original de cette méthode assure de bonnes performances en facilitant l'apprentissage et la satisfaction des utilisateurs mais il reste encombrant pour un dispositif mobile à petit écran. L'implémentation de cette méthode publiée dans [Martin, 2005] utilise un pavé tactile

externe et un affichage sur écran d'ordinateur de bureau. La combinaison de la surface d'interaction et de l'affichage (écran tactile sur PDA par exemple) est sans doute difficile à réaliser sans dégradation des performances.

3.2.7. TUP

TUP (Transparent User-guided Prediction) est une méthode qui profite de la molette tactile disponible sur certains dispositifs mobiles (iPod) et qui risque d'être plus largement utilisée avec le développement d'appareils comme les Smartphones et les assistants numériques. Le principe est d'afficher sur l'écran toutes les lettres organisées par ordre alphabétique autour d'un cercle (voir Figure I-13). Selon la position où l'utilisateur met son doigt sur la molette, une lettre est mise en surbrillance. Si la sélection n'est pas exacte, l'utilisateur peut tourner la molette pour sélectionner le caractère désiré. Si le mouvement de rotation est rapide, la sélection est ajustée par un système de prédiction de lettres dit « transparent » [Proschowsky et al., 2006].

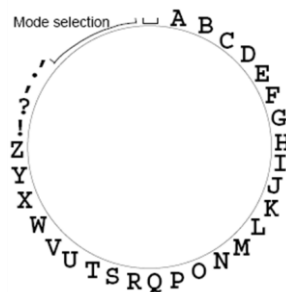


Figure I-12. Organisation des caractères autour de la molette tactile.

La méthode n'utilise pas une organisation des lettres statistiquement optimisée pour ne pas augmenter la charge mentale de l'utilisateur. L'évaluation empirique de cette méthode montre une saisie à 6,2 WPM avec seulement 1,2% d'erreurs. La méthode reste lente et utilise un grand espace sur l'écran du dispositif. L'impression des caractères autour de la molette a été évitée pour minimiser les allers-retours visuels entre la molette et l'affichage et pour éviter de figer la liste des caractères possibles.

3.2.8. Circumscript

Comme Claviature, Circumscript est un produit de Microth. Cette méthode de saisie a été commercialisée en 2006. Elle est basée sur un autre produit de la même firme : la « KeyWheel ». Cette dernière est une sorte de molette cliquable tactile. Un petit espace circulaire composé de quatre à six¹ touches tactiles permet à l'utilisateur d'effectuer des actions de défilements en traçant des arcs au doigt à travers ces touches. Le sens du mouvement du doigt (et ainsi du défilement) est détecté lors du passage du doigt entre deux touches voisines (voir Figure I-14).

Dans le cas de la méthode de saisie Circumscript, un ensemble de six à sept caractères est affecté à chacune des quatre touches de la KeyWheel. Les caractères

¹ La fonction « molette » est l'une des utilisations possibles de la KeyWheel. D'autres exemples, pour la saisie de texte entre autres, utilisent 4 ou 6 touches.

attribués à chaque touche sont divisés en deux groupes sur les deux cotés de la touche. Chaque caractère a un rang N de 1 à 4 dans le sous-groupe auquel il appartient. Pour saisir un caractère, l'utilisateur doit pointer (poser) le doigt sur la touche qui contient ce caractère, puis effectuer un geste rotatif dans le sens où se trouve ce dernier en passant par N touches (voir Figure I-14). Un feedback visuel permet à l'utilisateur de voir le caractère actif à chaque instant de son geste. Une fois que le doigt n'est plus en contact avec la KeyWheel, le dernier caractère actif est validé.

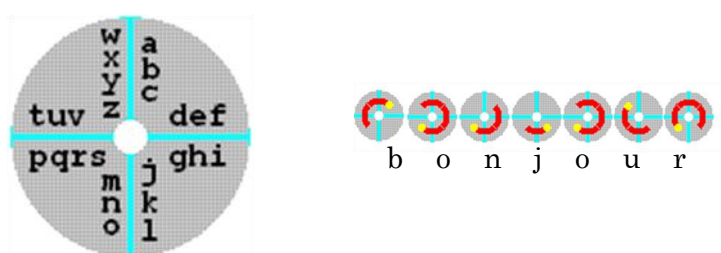


Figure I-13. La méthode de saisie Circumscript. Exemple de la saisie du mot « bonjour ».

3.2.9. EasyStroke

Développée aux laboratoires de l'université de Zhejiang à Hangzhou (Chine) en s'inspirant de la première version de Glyph (voir chapitre II), cette méthode s'appuie sur une performante API de reconnaissance des formes à accélération matérielle pour dispositifs mobiles (faibles ressources) appelée OpenVG [He et al., 2006]. Cinq primitives avec trois variations chacune ont été définies pour permettre de reconstruire analogiquement les lettres en dessinant en continu une séquence de 2 à 3 primitives (voir Figure I-15).

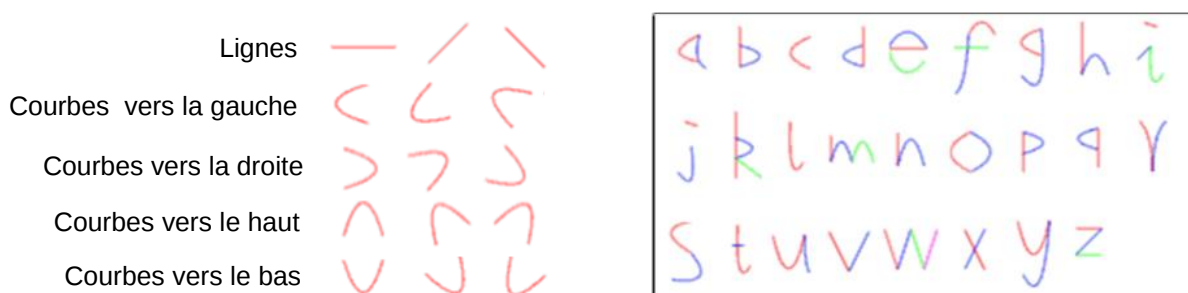


Figure I-14. La liste des primitives et la carte des caractères EasyStroke.

Ainsi, EasyStroke propose une meilleure analogie que celle de Glyph (voir chapitre II) en codant intégralement la forme des lettres, les points sur les caractères inclus (alors que Glyph code uniquement les formes caractéristiques de chaque lettre). Elle permet donc une saisie avec moins d'apprentissage. La méthode est mise à disposition comme logiciel libre pour Tablet PC sous Linux. Cette méthode reste très proche de la saisie manuscrite, elle est donc lente. Aucune étude des performances de la méthode n'a été publiée à ce jour. Il est à noter que la version pour PDA de Glyph 2 permet une saisie à un seul geste par caractère (1 GPC) même si les lettres sont codées par deux primitives (voir chapitre II).

3.2.10. FreePad

Les auteurs de cette méthode [Bharath et Madhvanath, 2008] partent des deux constats suivants :

- Les méthodes basées sur la reconnaissance des lettres sont trop lentes, entre autre, à cause de la temporisation (timeout) pour valider chaque caractère,
- Les méthodes utilisant une saisie continue nécessitent un grand espace et requièrent beaucoup d'attention de la part de l'utilisateur.

Pour remédier à ces deux problèmes, FreePad propose de fusionner les deux techniques en utilisant une petite surface d'écriture tout en permettant une écriture en continu. L'ordre conventionnel de l'écriture (par exemple : de gauche à droite pour les langues qui utilisent des caractères romains ou de droite à gauche pour les langues Abjadiques) est donc tout simplement ignoré pour superposer les lettres les unes sur les autres comme le montre la Figure I-16.



Figure I-15. Le principe de l'écriture avec FreePad.

Les expériences menées par Bharath et Madhvanath [2008], chercheurs de Hewlett-Packard India, montrent que FreePad atteint un taux de reconnaissance de 89,17% avec un dictionnaire de 20 000 mots et que le temps moyen de saisie est inférieur à celui observé avec un clavier T9. Aucune quantification des performances en vitesse de saisie n'est communiquée.

3.2.11. 7SI

7SI est une méthode de saisie gestuelle analogique pour interface tactile. Elle adopte une analogie mixte (certaines lettres en majuscule et certaines en minuscule) comme Graffiti™ et certaines autres méthodes. L'élément nouveau dans cette méthode est qu'elle s'appuie sur un alphabet et un feedback visuel composé de 7 « segments » comme les anciens afficheurs « 7-Seg Display ». Deux petits segments supplémentaires de forme circulaire ont été ajoutés au centre de chaque carré formé par 4 segments pour servir de raccourcis permettant de les activer tous les 4 en même temps. L'alphabet 7SI est présenté dans la Figure I-17.

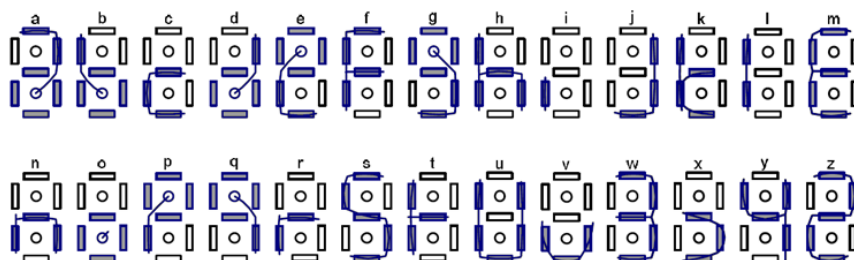


Figure I-16. L'alphabet 7SI.

L'analogie utilisée dans cette méthode réside surtout dans le feedback. Le geste correspondant à chaque caractère n'est pas forcément analogique. Cela permet de réduire significativement la longueur du geste pour certains caractères. Une étude prédictive basée sur la loi de Fitts estime la vitesse de saisie théorique maximale à 27,49 WPM [Al Faraj et al., 2008a]. Le principal inconvénient de cette méthode réside dans les dimensions de son interface actuelle qui occupe environ la moitié de l'écran du PDA. Il contient en plus du feedback 7-Seg, des boutons de fonctions diverses dont celles pour changer de mode. L'interface publiée dans [Al Faraj et al., 2008a] semble être un prototype et peut être améliorée.

Cette méthode de saisie peut être « indépendante du dispositif ». Il serait intéressant d'effectuer une évaluation avec des utilisateurs et d'étudier son implémentabilité sous différentes formes (clavier physique, clavier à accord ou manette par exemple). Une adaptation pour des handicaps moteurs légers existe déjà (voir section 4.2.1).

3.3. Méthodes à détection de l'inclinaison

3.3.1. Unigesture

Les puces de détection d'inclinaison (accéléromètres) se développent de plus en plus sur les appareils électroniques mobiles de petites dimensions. Cette technologie n'a pas échappé aux investigations des chercheurs de nouvelles solutions de saisie de texte. Dès l'année 2002, Roy Want de Intel Research et ses collègues de l'université de Washington ont développé plusieurs prototypes de deux méthodes de saisie utilisant cette technique.

La première s'appelle « Unigesture » et permet une saisie à une seule main [Sazawal et al., 2002]. Les lettres sont groupées en 7 groupes de 3 à 4 lettres associées, chacun correspond à une direction d'inclinaison (nord, nord-est, est...). Le caractère d'espacement est affecté, seul à la direction sud (vers le bas, voir Figure I-18-a). Pour saisir un mot, l'utilisateur effectue, pour chaque lettre l'inclinaison du dispositif qui correspond à son groupe. À la saisie d'un espacement, un système de désambiguïsation génère le premier mot correspondant à la séquence effectuée. Une touche « suivant » lui permet de choisir un autre mot dans la liste proposée. Une évaluation empirique de cette méthode comparant deux organisations des caractères a été effectuée avec 12 utilisateurs mais la vitesse de saisie n'a pas été communiquée [Sazawal et al., 2002].

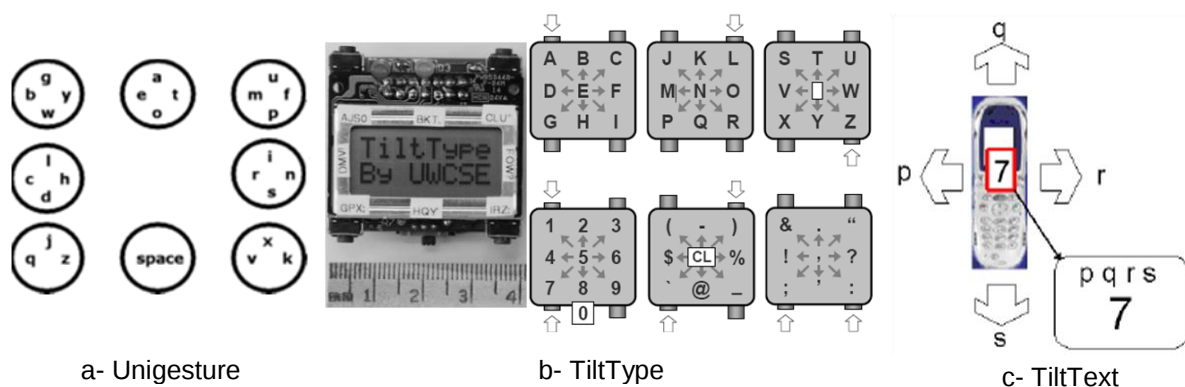


Figure I-17. Méthodes de saisie par détection de l'inclinaison.

3.3.2. *TiltType*

La deuxième méthode s'appelle « TiltType » (voir Figure I-18-b). Elle utilise quatre touches pour combiner plusieurs cartes de caractères avec neuf positions d'inclinaison : les huit directions (directions cardinales et diagonales) en plus de la position neutre (centre). La validation d'un caractère se fait au relâchement du ou des boutons appuyés.

3.3.3. *TiltText*

Une autre méthode développée sur le même principe par Wigdor et Balakrishnan [2003] s'appelle « TiltText ». Elle consiste à associer à chaque touche du clavier T12 d'un téléphone mobile quatre directions d'inclinaison pour permettre de saisir les lettres en un seul appui de touche contrairement à la méthode Multitap classique (voir Figure I-18-c). Une expérience menée avec 10 utilisateurs montre une vitesse de saisie de 13,57 WPM avec un taux d'erreurs de 11% après 16 sessions de test. Les inconvénients majeurs de ces méthodes restent le taux d'erreurs et la fatigue causée par la sollicitation motrice du poignet, de la main et du bras.

3.3.4. *Hex*

Publiée par des chercheurs de l'université de Glasgow la même année que VirHKey (présentée dans la section 3.2.6), cette méthode est venue avec la même idée de diviser les lettres en plusieurs groupes et d'utiliser une hiérarchie géométrique donnant un effet de zoom pour permettre une saisie gestuelle. La différence entre les deux méthodes réside dans la forme géométrique choisie, l'unité de saisie et le dégroupage des lettres :

- La forme géométrique des cellules : Hex utilise une géométrie hexagonale formant une grille régulière (mosaïque hexagonale) qui permet une navigation plane illimitée. Le plan est déformé par un effet de zoom pour désigner la maille active à chaque instant.
- L'unité de saisie : alors que l'unité de saisie de VirHKey est le caractère, Hex permet, en plus de la saisie des caractères une saisie continue de niveau mot grâce à une affectation dynamique des lettres aux mailles voisines de celle active. La saisie d'une lettre s'effectue en deux gestes, le premier pour accéder à l'hexagone correspondant à un groupe de lettres et le deuxième pour choisir l'une de ces lettres qui, après le premier geste, se repositionnent dans les hexagones voisins. Une fois la lettre désignée les groupes de lettres réapparaissent dans les hexagones voisins et le cycle recommence permettant une saisie continue.
- Les groupes de lettres : le groupage des lettres est fait de manière à optimiser le mouvement. L'un des groupes contient toutes les voyelles, l'espacement et la commande de suppression sont considérés comme des caractères et regroupés avec les lettres. Une distribution dynamique est aussi utilisée en se basant sur un modèle de prédiction de lettres [Williamson et Murray-Smith, 2005].

La Figure I-19 montre deux versions de Hex, l'une sur PC supportant la souris et l'accéléromètre et l'autre sur Pocket PC utilisant un petit accéléromètre. La version pour Pocket PC n'utilise pas l'effet de zoom. La grille reste illimitée et la saisie en continue reste possible.

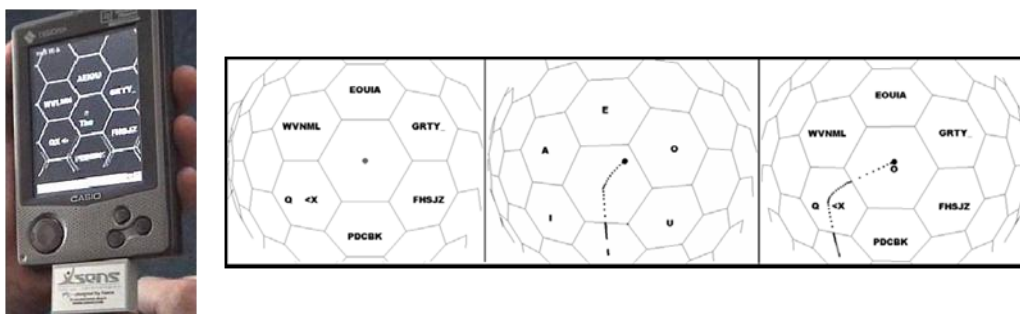


Figure I-18. L'interface de la méthode de saisie Hex. Une grille d'hexagones étendue à l'infini.

Concernant les performances, l'un des concepteurs de la méthode arrive à atteindre 10 à 12 WPM. Les auteurs annoncent que *des vitesses de l'ordre de 17 WPM sont atteignables*. Aucune évaluation théorique ni expérimentale formelle n'est publiée à ce sujet à ce jour. La méthode est originale et les vidéos de démonstration montrent une fluidité d'utilisation (par les concepteurs) mais elle reste inadaptée aux petits dispositifs mobiles vu son interface encombrante. Une étude sur les taux d'erreurs engendrées par cette méthode et sur la fatigue de l'utilisateur est nécessaire pour valider son efficacité.

3.3.5. UniGest

UniGest est une nouvelle méthode de saisie de texte de type unistroke qui utilise la télécommande de la console de jeu Wii® de Nintendo¹, la Wiimote, pour saisir du texte. Un alphabet analogique aux caractères romains en minuscules est proposé (voir Figure I-20) pour permettre de saisir chaque lettre en un ou deux mouvements simples appelés « primitives de mouvement ». Dix primitives sont utilisées au total. Dans le cas d'UniGest, on parle plutôt de « mouvement » car l'interaction se fait dans l'espace à la place d'une surface 2D utilisée habituellement avec le stylet. L'appui sur un bouton de la télécommande Wii délimite le mouvement comme le contact du stylet avec une surface tactile ou l'appui sur le bouton de la souris dans le cas d'un ordinateur de bureau.

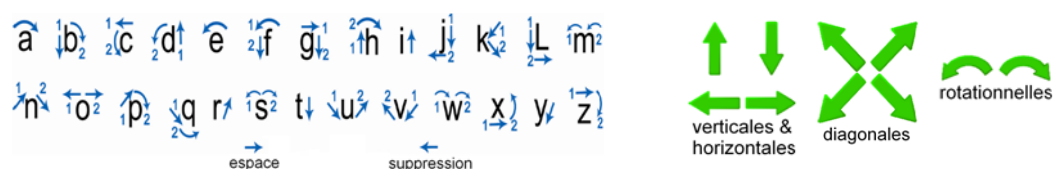


Figure I-19. L'alphabet UniGest et le jeu de primitives associé.

Une expérience a été menée en ligne avec des joueurs de la Wii® pour prédire la vitesse de saisie. Les participants ont été sollicités pour reproduire les gestes correspondants aux dix primitives sur une applet mise à disposition sur Internet. Aucune évaluation avec une tâche de saisie de texte n'a été publiée à ce jour. L'expérience en ligne a permis de prédire un WPM théorique de 27,9 [Castellucci et MacKenzie, 2008]. Cette valeur ne tient pas compte du temps nécessaire pour se rappeler les gestes correspondants aux caractères à saisir, ni du temps entre les gestes,

¹ Nintendo, site web europe : wii.nintendo-europe.com, visité le 29/01/2009.

ni du temps nécessaire pour remettre la main dans une position de repos après la saisie d'un caractère.

4. La saisie de texte pour personnes handicapées

Comme nous l'avons montré dans les sections précédentes de ce mémoire, la saisie de texte est devenue un des grands challenges de l'informatique mobile avec le développement et la diversité des appareils informatiques de petites dimensions. Les méthodes que nous avons présentées, visent en quelque sorte à remplacer le traditionnel clavier complet par des méthodes de saisie optimales soit physiques, soit virtuelles. Pour la plupart, celles-ci restent toutefois fort mal adaptées à la nature et à l'utilisation de ces dispositifs de petites dimensions. Il n'est donc pas envisageable de proposer aux personnes ayant un handicap physique quelconque d'utiliser ces mêmes méthodes et ces claviers miniatures disponibles sur les assistants personnels (PDA). Déjà pénibles pour les personnes valides, ces solutions sont totalement inutilisables par des personnes handicapées. Cependant, les dispositifs d'infocommunication modernes pourraient être aussi bénéfiques à cette population qu'aux personnes valides en leur assurant un plus grand degré d'autonomie en raison de leurs petites tailles, de leur facilité d'installation sur un fauteuil roulant, de leur faible consommation d'énergie et de leur faible temps d'initialisation et d'exploitation.

Dans cette section nous ne nous attarderons pas sur les études statistiques et la catégorisation des handicaps qui sont bien abordés dans de nombreux livres et rapports ([Ceccato et al., 2001] et [Brouard, 2004] par exemple) et sur des sites web gouvernementaux ou associatifs spécialisés¹. Nous étudierons uniquement les solutions pour la saisie de texte sur dispositifs mobiles proposées aux personnes handicapées en nous concentrant sur la saisie de texte avec un handicap moteur, sujet traité dans cette thèse.

4.1. Méthodes pour handicap visuel

Dans la littérature sur la saisie de texte, on trouve en premier lieu, des solutions pour les personnes atteintes d'un handicap visuel. Les dispositifs de saisie braille comme Aria Brail² ou Brailino³ par exemple sont trop lourds, grands et encombrants pour être utilisés dans un contexte mobile [Lagoá et al., 2007]. La méthode la plus basique, est le clavier téléphonique normal qui dispose d'un repère tactile sur la touche « 5 jkl ». Ce

¹ <http://www.handicap38.org/guidemetro-so.html>, par le Centre Départemental d'Information du Handicap de l'Isère, visité le 31/10/2008.

<http://www.doctissimo.fr/html/dossiers/handicap/handicap-handicape.htm>, par un portail de santé et de bien être, visité le 31/10/2008.

<http://www3.ac-clermont.fr/ASESH/spip.php?rubrique23>, par l'académie de Clermont-Ferrand, visité le 31/10/2008.

http://archives.handicap.gouv.fr/dossiers/handicaps/handicaps_diff1.htm, par un site web du ministère du travail, des relations sociales et de la solidarité, visité le 31/10/2008.

² Maxi Aids Inc., site web : <http://www.maxiaids.com/>, visité le 09/08/2008.

³ Handy Tech Elektronik, page web de Brailino : www.handytech.de/en/normal/products/for-blind/brailino/index.html, visitée le 09/08/2008.

repère permet à la personne mal voyante de localiser cette touche au milieu du clavier pour pouvoir ensuite déplacer le doigt sur les autres touches et effectuer les actions désirées. Une connaissance complète de la structure du clavier est nécessaire pour l'interaction. En outre, cette technique ne permet à l'utilisateur que de localiser les touches, aucun feedback sur le caractère saisi n'est fourni si ce n'est par l'utilisation d'un logiciel de lecture d'écran. Les méthodes les plus adaptées reposent surtout sur des retours haptiques et/ou sonores.

Parmi les méthodes les plus récentes, on cite LetterScroll [Tinwala et MacKenzie, 2008]. Cette méthode permet de saisir du texte en utilisant deux boutons et une molette. Une touche permet d'initialiser la position du curseur, la molette permet de déplacer le curseur sur une liste de lettres triées par ordre alphabétique et la deuxième touche permet de valider le mot en insérant un espacement. Un retour sonore permet à l'utilisateur de placer le curseur sur la lettre désirée. Des touches supplémentaires peuvent être utilisées pour optimiser le défilement en basculant directement sur l'une des voyelles. Cette méthode a été évaluée sur ordinateur de bureau avec des sujets aux yeux bandés en utilisant une souris à molette. Les résultats donnent une saisie à 3,6 WPM avec 2,68 à 6,97 KSPC selon l'implémentation [Tinwala et MacKenzie, 2008].

BloNo est une autre méthode qui utilise un curseur virtuel pour sélectionner les lettres. La différence avec LetterScroll réside dans l'organisation des lettres et l'outil de navigation et de sélection. Les lettres sont organisées alphabétiquement en cinq lignes commençant chacune par une voyelle. Le clavier téléphonique est utilisé comme une manette pour déplacer le curseur sur la matrice de lettres. Les touches '2', '8', '4' et '6' permettent de déplacer le curseur dans les quatre directions, un retour sonore permet de reconnaître la lettre sélectionnée à chaque déplacement, et la lettre est validée par un mécanisme de temporisation (comme dans un clavier multitap classique). La touche '9' permet d'accéder à une liste de mots commençant par le préfixe saisi et la touche '5' permet de valider le mot sélectionné [Lagoá et al., 2007]. La méthode a été évaluée avec des utilisateurs présentant une déficience visuelle totale. Après 5 minutes d'essai, ils ont réussi à saisir 2 mots par minute (WPM) avec 1,4% d'erreurs. Le taux d'erreurs a baissé à la fin de l'expérience pour atteindre 1% mais les performances finales en WPM et en KSPC ne sont pas communiquées explicitement.

4.2. Méthodes pour handicap moteur

Après les méthodes destinées aux personnes ayant des déficiences visuelles, on présente celles adaptées aux personnes ayant certaines déficiences motrices que nous qualifions de « légères » se manifestant sous forme de tremblement ou de contrôle réduit des mouvements. On peut diviser ces méthodes en deux grandes catégories : les méthodes analogiques et les méthodes à désignation de cibles.

4.2.1. Saisie analogique

L'une des techniques utilisées dans ce domaine est le guidage tangible. Cette technique est aussi proposée dans le domaine de l'interaction en véhicules automobiles [Kamp, 1998]. En entrée de texte adaptée au handicap moteur, la principale méthode qui utilise cette technique se nomme EdgeWrite™. Elle propose un alphabet analogique

(comme Graffiti™ par exemple) basé sur la saisie au stylet dans une petite surface tactile de forme carrée délimitée par des bordures physiques en relief (voir Figure I-21). Ces bordures physiques permettent de guider le stylet et faciliter l'interaction des personnes ayant des difficultés motrices. Chaque caractère peut être saisi de différentes façons selon une analogie mixte aux caractères en majuscules et en minuscules (voir annexe 1). Cela permet de réduire les erreurs puisque l'utilisateur peut utiliser l'analogie qui lui semble la plus logique et qu'il arrive le mieux à retenir mais ce choix pose aussi des problèmes de confusions dans le cas des caractères qui n'ont qu'une seule représentation. L'un des avantages d'EdgeWrite™ est que tous les caractères peuvent être saisis dans le même petit espace carré et sans changer de mode mais l'avantage principal de cette méthode reste son indépendance au dispositif. Plusieurs implémentations avec des dispositifs différents ont été présentées depuis le dépôt du brevet en 2004¹.

EdgeWrite™ a été adaptée en effet pour une utilisation sur chaise roulante [Wobbrock et al., 2004b ; Myers et Wobbrock, 2005 ; Wobbrock et al., 2005]. Le dispositif de contrôle de la chaise est utilisé dans cette implémentation pour saisir du texte. Deux types de dispositifs ont été testés : la manette directionnelle et le pavé tactile. Une évaluation de ces implémentations avec 6 personnes avec infirmités motrices cérébrales (IMC) et une personne avec une sclérose en plaques² montre que les utilisateurs arrivent à atteindre 0,77 WPM avec 29,56% d'erreurs en utilisant la manette et 1 WPM avec 25,40% d'erreurs en utilisant le pavé tactile [Wobbrock et al., 2004b].

Une autre implémentation d'EdgeWrite sur téléphone mobile (voir Figure I-21) utilise une petite manette intégrée à l'appareil. On trouve ce genre de manette sur de nombreux téléphones mobiles du marché. La manette est généralement positionnée entre l'écran et les touches de façon à être manœuvrée par le pouce mais Wobbrock et ses collègues [Chau et al., 2006] [Wobbrock et al., 2007] proposent en plus de cette disposition une intégration de la manette à l'arrière de l'appareil. La manette est disposée en partie supérieure à l'arrière de l'appareil à un quart de la largeur de ce dernier à partir de la droite de façon à être manœuvrée par l'index. Ces implémentations ne sont pas adaptées aux personnes handicapées mais illustrent l'indépendance du dispositif (versatility) de cette méthode et la possibilité de son utilisation par des personnes valides comme par des personnes handicapées.

¹ EdgeWrite a été publiée pour la première fois en 2003 [Wobbrock, 2003], un brevet US a été déposé en Mars 2004 par Jacob O. Wobbrock et Brad A. Myers sous le numéro 10/811,761.

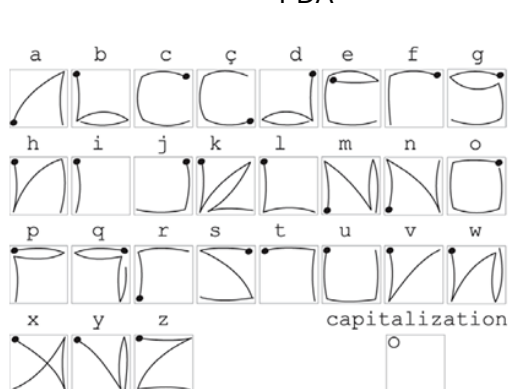
² Pour des informations sur les différentes déficiences motrices, voir le Dictionnaire du handicap [Zribi et Poupée-Fontaine, 2007].



Implémentation d'EdgeWrite™ sur PDA



Implémentation d'EdgeWrite™ avec molette.



Carte des caractères EdgeWrite™ (simplifiée)



Implémentation EdgeWrite™ à 4 boutons



EdgeWrite™ sur téléphone mobile avec manette isométrique

Figure I-20. Carte de caractères simplifiée et différentes implémentations d'EdgeWrite.

EdgeWrite™ a été implémentée aussi pour une saisie adaptée à molette [Wobbrock et Myers, 2006b]. La solution est nommée « Trackball EdgeWrite ». Elle permet de reproduire les caractères EdgeWrite à l'aide d'une boule de commande (aussi appelée boule roulante). C'est un dispositif de pointage très proche de la souris à bille, à la différence que l'utilisateur bouge la bille et non pas tout le dispositif. Une évaluation théorique selon la loi de Fitts de la méthode donne une vitesse de saisie de 23,1 WPM. L'évaluation expérimentale avec des utilisateurs valides donne une vitesse de 9,82 WPM avec un taux d'erreurs de 3,90% et une vitesse de 5,28 WPM avec 11,80% d'erreurs par un utilisateur à capacités motrices réduites à cause d'une lésion médullaire. D'autres méthodes comme MDITM [Isokoski et Raisamo, 2000] permettent une utilisation avec molette directionnelle ou avec boule roulante mais aucune évaluation avec des personnes handicapées n'est publiée à ce jour.

Une autre implémentation d'EdgeWrite utilise quatre touches pour reproduire un caractère en effectuant des appuis représentant les extrémités et les angles de son dessin (voir Figure I-21). L'évaluation de cette implémentation avec des utilisateurs valides montre une vitesse de 12,50 WPM avec un taux d'erreurs de 8,92%.

Toujours dans les méthodes analogiques, on trouve parmi les plus récentes GroovWrite [Al Faraj et al, 2008b]. Elle est basée sur le même principe de la méthode 7SI (présentée dans la section 3.2.11). Pour l'adaptation au handicap moteur un guidage physique avec des rainures est adopté pour stabiliser le mouvement du stylet. La différence des rainures avec les bordures d'EdgeWrite est que le stylet est guidé des deux

cotés. Une prédiction de la vitesse de saisie selon la loi de Fitts donne 24,78 WPM. Aucune évaluation empirique n'a été publiée à ce jour.

4.2.2. Saisie par désignation de cibles

Comme pour les personnes avec des déficiences visuelles, il existe des claviers complets adaptés aux handicaps moteurs. Les claviers Access¹ (Medi et Maxi) par exemple offrent un guidage physique (tangible) du doigt avec des bordures autour des touches. Ces claviers sont encombrants et ne peuvent être utilisés avec des dispositifs d'infocommunication mobiles.

Une autre approche consiste à utiliser une manette pour la saisie de texte en sélectionnant un item dans une liste ou dans une matrice. La méthode Date Stamp familière aux amateurs des jeux vidéo (souvent utilisée pour entrer les pseudos des joueurs) permet une saisie à la manette [Tarasewich, 2003] ou en utilisant trois boutons [MacKenzie, 2002], deux pour positionner le curseur sur une liste de caractères et le troisième pour valider. Cette méthode peut être utilisée à la manette par les personnes ayant un handicap moteur mais elle est très lente. Une évaluation de Date Stamp à la manette avec des utilisateurs valides donne une vitesse moyenne de 4,43 WPM [Wobbrock et al., 2004a]. Les claviers logiciels peuvent aussi être utilisés avec une manette notamment dans les jeux vidéo [Wobbrock et al., 2004a]. Wobbrock et al., [2004b] ont évalué l'utilisation de la manette avec le clavier WiVik [Shein et al., 1991] par des utilisateurs handicapés moteurs. Les résultats donnent une vitesse de 0,84 WPM avec un taux d'erreurs de 5,11%.

Weegie² est une autre méthode de saisie par désignation de cibles à la manette. L'utilisateur doit tourner la manette dans différentes directions avant de retourner au centre pour saisir un caractère (par exemple du centre vers 9 heures, puis retour au centre pour saisir la lettre 'a' ; du centre vers 6 heures, puis tourner dans le sens inverse aux aiguilles d'une montre jusqu'à 12 heures et revenir au centre pour saisir la lettre 'b', voir Figure I-22). KeyStick [Bloor, 2003], est aussi une méthode de saisie à manette pour une utilisation sur certains téléphones mobiles. Elle permet de sélectionner les caractères dans un menu avec des mouvements dans les quatre directions : droite, gauche, haut et bas. Aucune étude scientifique n'a été publiée sur ces deux méthodes.

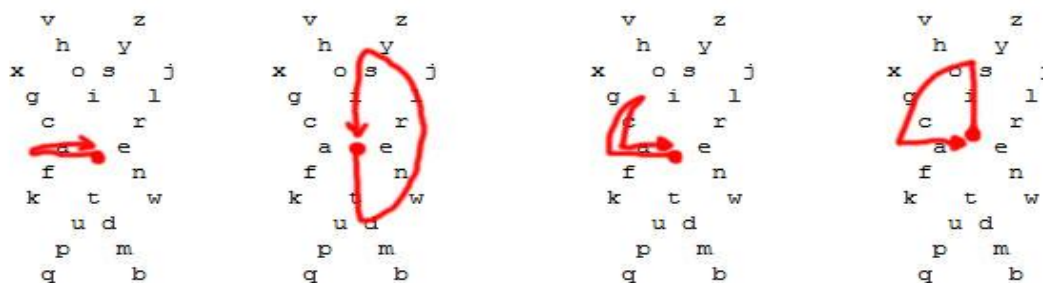


Figure I-21. Weegie : exemples de saisie des lettres 'a', 'b', 'c' et 'h' (respectivement de gauche à droite).

¹ Zygo Industries, pag web : <http://www.zygo-usa.com/>, visité le 12/08/2008.

² Weegie pour X11, disponible à : <http://weegie.sourceforge.net>, visitée le 12/08/2008.

Dans la catégorie des méthodes de saisie par désignation de cibles, on peut classer aussi Dasher [Ward et al., 2000], une méthode qui peut être utilisée avec tout type de dispositif de pointage (souris, boule roulante, manette, détection du regard...). Elle consiste à déplacer verticalement un pointeur sur l'écran pour cibler les lettres qui correspondent à la position horizontale du curseur sur la droite de l'écran. Les lettres validées passent sur la partie gauche de l'écran pour constituer le texte saisi (voir Figure I-23). Les lettres générées sur la droite sont prédites à partir du préfixe validé à gauche. Le déplacement du curseur horizontalement permet d'accélérer le zoom sur les caractères et de sélectionner directement une succession de caractères qui constituent la fin du mot en cours de saisie. Cette méthode nécessite un grand espace d'affichage et occupe l'intégralité de l'écran sur un dispositif mobile. Elle peut bien être utilisée par des personnes handicapées mais, à notre connaissance, aucune évaluation avec des personnes handicapées moteur n'a été publiée à ce jour.

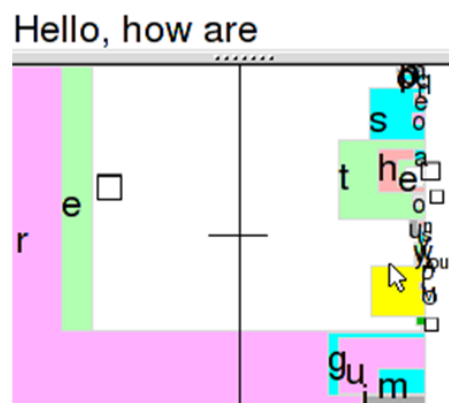


Figure I-22. Dasher, les dernières lettres sélectionnées sont 'r', 'e' et espace.

En plus de l'utilisation des manettes, des touches ou encore de la détection du regard pour la désignation de cibles, d'autres techniques se développent pour permettre à la personne handicapée d'agir sur un dispositif informatique. Hex-O-Spell par exemple utilise une interface cerveau-ordinateur pour la saisie de texte [Blankertz et al., 2006]. Cette méthode est basée sur la méthode Hex présentée dans la section 3.3.4. Deux états de contrôle simples sont exploités pour explorer les hexagones. Imaginer un mouvement de la main droite permet de tourner un pointeur (flèche) au centre du clavier. Imaginer un mouvement du pied droit permet de stopper la rotation et rallonger le pointeur pour sélectionner l'hexagone ciblé (voir Figure I-24). Chaque caractère est saisi en deux étapes, la sélection du groupe auquel il appartient puis la sélection du caractère lui-même.

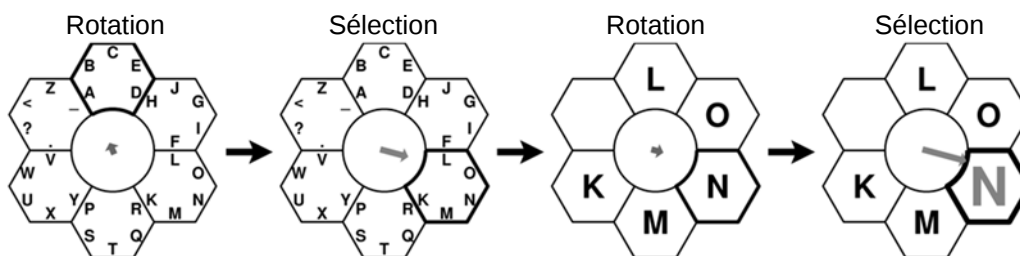


Figure I-23. Hex-O-Spell, saisie avec une interface cerveau-ordinateur. Exemple de la saisie de la lettre 'N'.

Deux personnes ont testé le mécanisme et ont réussi à atteindre des vitesses de saisie respectives de 1 WPM et 1,52 WPM.

4.3. Systèmes de communication palliative

De nombreux systèmes d'aide à la communication (appelés aussi systèmes de communication palliative ou AAC pour Augmentative and Alternative Communication) pour les personnes ayant des déficiences motrices sont disponibles sur le marché (par exemple : SwitchXS™, commercialisé par AssistiveWare¹ ou ClaviComNG², développé par Thomas Devaux et Guillaume Rebesché dans le cadre d'un projet du programme France, de Handicap International³ et mis à disposition sous licence libre GNU GPL). D'autres sont en cours de développement ou d'amélioration dans certains laboratoires de recherche (Sibylle [Wandmacher et al, 2007] par exemple). Ces systèmes visent à permettre à la personne handicapée de communiquer oralement et par écrit. Ils sont généralement constitués d'une méthode d'entrée de texte et de commandes et d'un module de synthèse vocale. Dans la suite de cette section, nous mettons l'accent sur la partie « saisie de texte » en étudiant les différentes techniques mises en œuvre qui peuvent enrichir notre travail.

Les méthodes de saisie de texte utilisées dans les systèmes d'aide à la communication peuvent être classées en deux grandes catégories :

- les méthodes iconiques telles que MinSpeak [Chang et al., 1992] et PCA Icon [Bellengier et al., 2004] où un pictogramme correspond à un mot, une phrase ou un concept.
- les méthodes linguistiques telles que WiViK [Smith et al., 1996], VITIPI [Boisière et Dours 2000] et Sibylle [Wandmacher et al, 2007] qui permettent de saisir du texte avec un nombre d'appuis par caractère (KSPC) minimal et une économie d'appuis (KSR) maximale. Différents principes de prédiction sont exploités (prédiction de lettres, de mots, de phrases, prédiction lexicale, syntaxique, sémantique, conceptuelle ou contextuelle ...).

Outre la prédiction, des techniques de défilement et de balayage [Cantegrit et Toulotte 2001] sont utilisées dans les cas d'utilisateurs ayant des capacités motrices très limitées qui ne peuvent effectuer qu'un simple mouvement. Ce mouvement basique permet d'actionner un contacteur binaire. Dans ces cas, la sélection est assurée par le système, contrairement aux méthodes qu'on a décrit précédemment où l'utilisateur sélectionne lui-même une touche en utilisant une manette, des touches ou un autre dispositif. Il s'agit généralement de mettre séquentiellement en surbrillance les objets (notamment les caractères) composant le clavier. L'action de l'utilisateur sur le

¹ Assistiveware Inc., page web de SwitchXS : www.assistiveware.com/switchaccess.php, visitée le 12/08/2008.

² Handicap-Icom, page web de ClavicomNG : www.handicap-icom.asso.fr/adaptations/aides_techniques/clavicomtech.html, visitée le 12/08/2008.

³ Programme France – Handicap International, page web : www.handicap-international.fr/en-france/le-programme-france/, visitée le 02/11/2008.

contacteur permet de valider l'objet sélectionné. Des techniques de balayage ligne-colonne/colonne-ligne ou de balayage dichotomique par exemple sont également utilisées pour augmenter la vitesse d'accès à un item sur les claviers comportant un grand nombre de touches. Plusieurs types de contacteurs avec des tailles, des sensibilités et des fixations différentes sont disponibles sur le marché selon les types des handicaps et les possibilités motrices des utilisateurs.

La plupart de ces systèmes d'aide à la communication existants sont conçus pour les ordinateurs de bureau et les ordinateurs portables (MacBook, PC portable, Tablet PC...). Peu de travaux de recherche ont abordé directement le sujet de la saisie de texte sur dispositifs mobiles pour personnes ayant un handicap moteur lourd. Ces dispositifs présentent beaucoup de limitations matérielles et techniques (petit écran, faible mémoire, processeur moins rapide...) qui ne permettent pas d'adapter directement les systèmes de communication palliative déjà existants sur ordinateur de bureau à une utilisation nomade. KeCom¹ est l'un des rares systèmes d'aide à la communication sur PDA à l'heure actuelle. Il a été développé au sein du Centre Mutualiste de Rééducation et de Réadaptation Fonctionnelles de Kerpape (CMRRF) par Jean-Paul Departe, ingénieur au laboratoire d'électronique du centre.

KeCom est composé d'une petite zone d'édition, d'un clavier de cinq lignes et neuf colonnes et d'une matrice de neuf touches qui affichent les mots prédits en mode « saisie » et des commandes pour lancer la synthèse vocale, envoyer le texte par SMS ou encore effacer la zone d'édition en mode de commande. Un curseur parcourt séquentiellement les touches de la zone active (clavier/zone de prédictions et de commandes). L'utilisateur peut effectuer à tout moment un appui sur un bouton du pavé directionnel du PDA, sur l'écran du PDA ou sur un contacteur externe branché à l'appareil pour valider la touche sélectionnée. Le clavier réorganise dynamiquement les lettres de gauche à droite et de haut en bas selon leurs probabilités d'apparition en fonction du préfixe déjà saisi. Les deux premières touches ne sont pas des touches de caractères, elles assurent certaines fonctions comme le basculement du curseur de défilement entre le clavier et la zone de prédiction ou de commandes. Trois niveaux d'appuis sont utilisés par KeCom pour permettre plusieurs fonctions avec la même touche : appuis simples, appuis longs et appuis très longs. Quand le caractère à saisir n'est pas sur la première ligne, l'utilisateur peut soit attendre que le curseur y arrive normalement en passant par tous les caractères qui le précèdent, soit basculer à l'aide de la première touche du clavier '@' à un balayage ligne-colonne. Dans ce cas, le curseur défile sur les touches de la première colonne (chaque bouton de la première colonne permet de sélectionner une ligne), l'utilisateur valide ensuite pour que le curseur défile sur les touches de la ligne sélectionnée.

¹ Page web : www.kecom.info, visitée le 13/08/2008.



Figure I-24. KeCom, pour la saisie de texte à un seul bouton poussoir sur PDA.

KeCom est utilisé et apprécié par plusieurs personnes au CMRRF de Kerpape. Aucune évaluation scientifique n'est publiée à ce jour.

5. Conclusion

Dans ce chapitre, nous avons examiné un certain nombre de métriques de saisie de texte et les travaux de recherche qui les ont étudiés. Nous avons survolé les métriques relatives à la vitesse de la saisie et à l'effort fourni par l'utilisateur lors de l'interaction. Ces mesures ne sont pas toutes communiquées pour toutes les méthodes.

Nous avons fait aussi une étude des principales méthodes de saisie de texte pour dispositifs informatiques mobiles qui ont apparues entre 2005 et 2008 et cité les plus célèbres. L'étude montre une grande variété de techniques utilisées. La classification ainsi que la comparaison des performances de chacune de ces méthodes sont difficilement réalisables. La plupart des valeurs rapportées dans ce chapitre correspondent aux performances maximales publiées. L'un des plus importants travaux comparatifs de l'état de l'art a été fait par Benoît Martin et Isabelle Pecci [2007] en recueillant toutes les performances théoriques et empiriques connues d'une soixantaine de méthodes différentes dans deux tableaux, l'un pour les méthodes à touches et l'autre pour les méthodes gestuelles. L'étude des méthodes que nous avons présentées montre, comme beaucoup de travaux d'état de l'art, la difficulté de comparer ces méthodes mais aussi l'absence d'une méthode de saisie « totalement » satisfaisante.

Concernant les méthodes de saisie dédiée aux personnes handicapées on peut retenir plusieurs techniques pour rendre un appareil informatique accessible et utilisable par cette catégorie d'utilisateurs. La saisie à l'aide d'une manette est l'une des techniques les plus référencées. La réduction du nombre de touches, l'utilisation des contacteurs, de la prédiction et du balayage sont aussi des techniques qui facilitent la saisie de texte aux personnes à capacités motrices très réduites. Le domaine de la saisie sur dispositifs mobiles adaptée aux handicaps moteurs lourds est très peu exploré. Ces dispositifs peuvent, pourtant, être d'une très grande utilité. Pour améliorer l'autonomie de ces personnes et leur permettre d'accéder aux avantages (compacité, légèreté, autonomie) et aux services de l'informatique mobile, il n'y a pas d'autre voie que de concevoir des

méthodes de saisie adaptées à leurs handicaps. La « saisie analogique par décomposition » est une technique de saisie qui peut être efficace sur les dispositifs mobiles et pour les personnes handicapées car elle permet d'avoir des claviers à très peu de touches. Dans la suite, nous étudions les méthodes existantes qui se basent sur cette technique.

CHAPITRE II

LA SAISIE ANALOGIQUE À PRIMITIVES

Résumé

La décomposition des caractères en formes de base selon un principe d'analogie de forme est l'une des techniques qui permettent de réduire considérablement le nombre de touches tout en assurant une facilité d'apprentissage. Dans ce chapitre, nous étudions dans un premier temps les principales méthodes de saisie basées sur cette technique, puis nous décrivons nos travaux sur le projet GLYPH qui a servi de précurseur à cette thèse.

1. Introduction

La décomposition des caractères en formes de base selon un principe d'analogie de forme avec les caractères romains est l'une des techniques qui permettent de réduire le nombre de touches utilisées pour la saisie de texte sur dispositifs mobiles. Un second principe est souvent respecté dans le codage des caractères. Il consiste à suivre le sens du tracé (ductus) ; formellement, il correspond à une analogie temporelle avec l'écriture en script du caractère. Par exemple, l'écriture de la lettre 'b' commence par le trait vertical de haut en bas, celle du 'x' commence par la partie gauche. Ces deux principes d'analogie spatiale et temporelle favorisent grandement la mémorisation d'un codage par décomposition des caractères en traits caractéristiques de leurs tracés.

Dans ce chapitre, nous présentons quelques méthodes qui utilisent ces techniques en mettant l'accent sur Glyph, le prédécesseur de la méthode UniGlyph que nous détaillerons dans les chapitres suivants. Dans la littérature, on trouve plusieurs appellations des segments graphiques d'un caractère (segments, fragments, traits, primitives...etc). Nous utiliserons l'appellation « primitives » introduite par Hsu dans [Hsu, 1991].

2. Méthodes existantes

Les premières méthodes qui utilisent la technique de décomposition des caractères en primitives sont destinées aux langues asiatiques de l'est comme le Chinois [Chen et Gong, 1984 ; Hsu, 1991] vu que le nombre des caractères est très élevé et que chaque caractère (idéogramme) est à l'origine composé d'un ensemble de dessins de base qui correspondent à des pictogrammes (dessin simplifié d'une chose ou d'un objet basique dans la nature). Ces pictogrammes sont eux aussi composés de formes de bases (traits, carrés, cercles ...). Dans les autres langues, surtout celles basées sur les caractères

romains, cette technique est essentiellement adoptée pour réduire le nombre des touches pour la saisie de texte.

Le principe de la décomposition des caractères a été ensuite adopté par quelques méthodes de saisie destinées à des langues occidentales. Nous décrivons dans la suite les principales méthodes qui utilisent ce principe.

2.1. Symbol Creator

Symbol Creator [Luoma, 2003] est une méthode de saisie de texte analogique à décomposition des caractères pour entrées à touches physiques ou à touches virtuelles ainsi que pour des entrées gestuelles. Le principe de Symbol Creator est de décomposer le dessin manuscrit en minuscule de chaque lettre de l'alphabet en un maximum de trois primitives (appelées « segments ») parmi un ensemble de sept primitives de formes et une primitive de contrôle « END » (voir Figure II-1). Cette dernière sert à la segmentation afin de définir la fin de la saisie pour les caractères composés d'une seule primitive ('o' et 'i' par exemple).

Une implémentation de Symbol Creator pour une entrée tactile texturée est très proche des techniques de reconnaissance des lettres manuscrites. L'apport est de faciliter la reconnaissance, tout d'abord en diminuant le nombre de formes à reconnaître par le système, puis en utilisant un mécanisme de reconnaissance basé sur le traitement des séquences de fréquences sonores résultant du mouvement du stylet sur une surface texturée [Evreinov et Raisamo 2003]. Chaque segment est caractérisé par une fréquence distincte (voir Figure II-1).

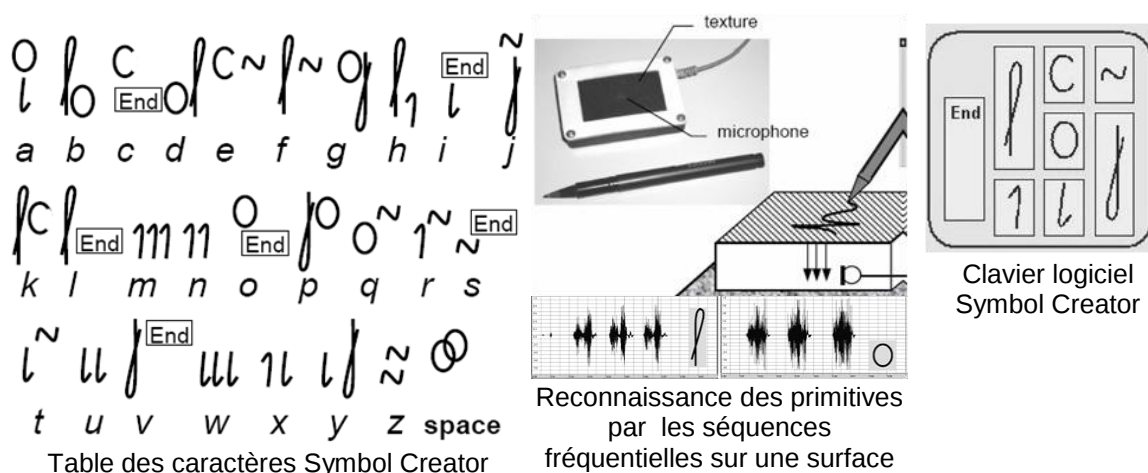


Figure II-1. Symbol Creator, une méthode de saisie analogique par décomposition des caractères.

D'autres implémentations de Symbol Creator ont été faites surtout sous forme de claviers virtuels dont l'un [Miniotas et al., 2003] est dédié aux personnes avec des handicaps moteurs. Cette implémentation consiste à sélectionner (non pas dessiner) les primitives par détection du regard. En effet, avec un clavier virtuel complet, il est difficile de sélectionner efficacement les touches par détection de la direction du regard. Le faible nombre de primitives rend cette tâche considérablement plus facile.

Les évaluations empiriques publiées par Evreinov et ses collègues ainsi que celles faites par Melinda Luoma [2003] montrent que les utilisateurs novices arrivent à saisir 8,58 mots par minute sur le clavier à détection du regard et 11,1 WPM sur un clavier logiciel Symbol Creator à huit touches implémenté sur un dispositif mobile à écran tactile de 6,5". Symbol Creator représente cependant deux inconvénients majeurs malgré son principe d'analogie assez claire :

- Les primitives utilisées ne permettent pas de représenter les chiffres et les caractères spéciaux même en combinant plus de deux.
- L'analogie pour certains caractères alphabétiques comme le 'x' et le 'z' est très difficile (ou les combinaisons sont mal choisies).

2.2. HandyScript

D'autres méthodes de saisie adoptent les techniques de l'analogie et la décomposition tout en proposant une carte de caractères plus large. HandyScript [Shu, 2003] par exemple utilise un plagiat du pavé numérique du clavier de l'ordinateur de bureau en associant 11 primitives aux touches numériques en plus de la touche de la virgule pour permettre de reconstruire analogiquement les caractères en combinant plusieurs primitives. Elle propose deux « niveaux » de saisie :

- Le niveau dit « vert » : permet de saisir une lettre en introduisant toutes les primitives qui le constituent dans l'ordre spatio-temporel de l'écriture. Une lettre peut être composée de une à quatre primitives. La segmentation est effectuée à l'aide de la touche « enter » ce qui donne entre 2 et 5 clics pour saisir chaque lettre (voir Figure II-2). L'utilisateur peut aussi dès l'entrée de la première primitive, faire défiler les choix possibles en utilisant les touches « + » et « - » puis valider avec la touche « enter ».
- Le niveau dit « rouge » : permet de saisir toutes les lettres en, exactement, deux primitives. La segmentation dans ce cas est automatique puisque la longueur des lettres est uniforme mais l'analogie perd de sa précision.



Exemples de décompositions
du niveau vert.



Exemples de décompositions du
niveau rouge.



Le clavier HandyScript

Figure II-2. HandyScript, une méthode analogique à onze primitives.

Les primitives de HandyScript, contrairement à celles de Symbol Creator, permettent de saisir les lettres avec une analogie à leurs formes en majuscules. Elles permettent

aussi de saisir un ensemble de caractères spéciaux dont les signes de ponctuation. Un changement de mode de saisie est nécessaire et c'est la touche « Num Lock » qui assure cette fonction. HandyScript est aussi disponible dans d'autres langues que celles basées sur les caractères romains et elle utilise le même jeu de primitives pour toutes les langues.

À notre connaissance, aucune étude scientifique sur les performances de cette méthode n'a été publiée. Son désavantage principal reste le nombre important de primitives et donc de touches.

2.3. Neopad

Neopad est une autre méthode de saisie analogique à décomposition. Elle est commercialisée par Neopad Inc.¹ depuis 2001 pour Palm. Actuellement, cette méthode est disponible dans plusieurs langues utilisant différents scripts dont le coréen, le chinois et l'arabe. Dix primitives sont utilisées au total. Pour les langues basées sur les caractères romains, l'analogie respectée est celle des lettres imprimées en majuscules. Chaque lettre est saisie par une séquence de deux primitives (voir Figure II-2). Le clavier contient, en plus des touches de primitives, différentes touches de fonctions et de commandes. Plusieurs modes de saisies sont mis à disposition de l'utilisateur. Dans le cas du clavier logiciel Neopad pour Palm, les caractères des modes « chiffres » et « ponctuation » sont directement affichés sur les touches et saisissables en un seul appui. Une implémentation de Neopad sur claviers téléphoniques est aussi prévue et annoncée dans les manuels d'utilisateur distribué avec l'application. Aucune étude des performances de ces claviers n'est déjà effectuée. Le nombre de touches reste élevé et la surface occupée par le clavier Neopad sur l'écran du Palm reste très proche de celle occupée par un clavier logiciel complet.

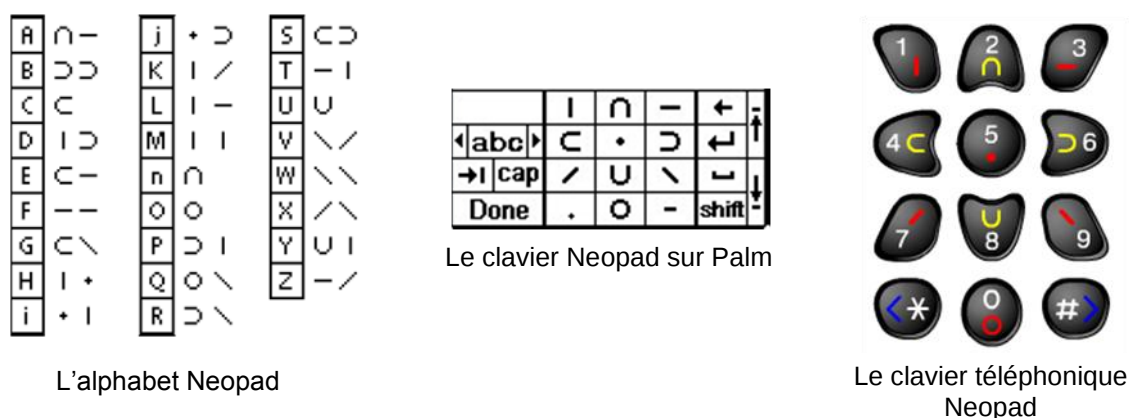


Figure II-3. Neopad, une méthode analogique à dix primitives.

D'autres méthodes utilisent ces techniques d'analogie avec décomposition mais l'inconvénient principal est la difficulté de trouver un jeu de primitives réduits qui permet de reproduire fidèlement un grand nombre de caractères. Le nombre d'appuis par

¹ NEOPAD Inc. page web : www.neopad.com, visitée le 14/08/2008.

caractère (KSPC) est aussi élevé en comparaison avec un clavier complet (dont le KSPC est proche de 1). Les méthodes les plus récentes qui ont adopté l'analogie avec décomposition des caractères en primitives comme EasyStroke, 7SI et UniGest qui ont été présentées dans le chapitre précédent, ont opté pour la saisie gestuelle pour surmonter ces problèmes.

Dans la suite, nous présentons une méthode de saisie pour dispositifs mobiles qui utilise moins de primitives pour occuper moins d'espace tout en offrant une large table de caractères et un effort minimal lors de la saisie.

3. Glyph, une saisie à six primitives

Lancé en 2004, le projet GLYPH¹ vise à construire des méthodes de saisie performantes et indépendantes du dispositif. Les techniques de l'analogie et de la décomposition ont été retenues pour les possibilités de réduire considérablement la taille du clavier tout en s'appuyant sur la connaissance préalable des formes des lettres par tout utilisateur. Les premières versions de Glyph ont été développées par Gurvan Uguen et Franck Poirier en 2004/2005 [Uguen et Poirier, 2004] [Poirier, 2005]. Ensuite la méthode a été améliorée avec Glyph2 par Franck Poirier et Mohammed Belatar [Poirier et Belatar, 2006a] dans le cadre d'un projet de Master2 et elle a été finalisée ensuite dans cette thèse [Belatar et Poirier, 2007a].

Glyph repose sur une analogie aux caractères romains en minuscules dans leur typographie la plus simple. Cette analogie a été choisie pour deux raisons :

- Du point de vue pratique : pour écrire un texte long ou une simple note, on utilise généralement des caractères en minuscules. Les majuscules étant uniquement utilisées pour les règles de capitalisation (début de phrase, nom propre, etc.), pour écrire un mot n'appartenant pas à la langue, pour mettre en valeur un mot ou pour qu'il soit mieux identifiable (comme le mot ATTENTION pour mettre en garde par exemple).
- Du point de vue technique : la décomposition des lettres en majuscules n'a pas permis d'avoir un jeu de primitives minimal, permettant de composer la totalité des caractères alphanumériques avec des combinaisons minimales. La majorité des méthodes existantes ont utilisé l'analogie Majuscule sans pouvoir atteindre un grand niveau d'optimisation.

En utilisant cette analogie aux minuscules, la décomposition des caractères donne un jeu de six primitives qui sont décrites dans le Tableau II-1.

¹ GLYPH ici est le nom du projet. Son origine est le mot *glyph* en anglais ou *glyphe* en français qui signifie :

- Selon Le Petit Robert : trait gravé en creux ;
- Selon Encarta : signe d'une ancienne écriture (maya ou chinoise) ; abstrait ornemental gravé en creux.
- Selon Oxford Dictionary : a pictograph or sculptured symbol; a small graphic symbol.

Nom de la primitive	Forme
Incurvée droite (ID)	
Incurvée gauche (IG)	
Incurvée haut (IH)	
Incurvée bas (IB)	
Horizontale (H)	
Verticale (V)	

Tableau II-1. Les six primitives de formes dans la décomposition Glyph.

Ces primitives sont combinées en séquences de trois au maximum pour chaque caractère. Ainsi, on peut coder avec ces six primitives un alphabet de $6^3 + 6^2 + 6 = 258$ caractères différents. Une septième primitive est utilisée pour la segmentation des séquences de primitives et pour d'autres fonctions. La segmentation peut aussi être effectuée par temporisateur dans les cas de claviers Glyph à touches ou au levé du stylet dans les cas d'entrée gestuelle (voir les implémentations de Glyph dans la suite). La saisie se fait en introduisant la séquence de primitives correspondantes à un caractère dans l'ordre spatio-temporel de son écriture à la main. Le Tableau II-2 montre les exemples de la décomposition des lettres 'g' et 'h' et l'ordre d'entrée des primitives. Dans le cas où la lettre est codée par une ou deux primitives, la touche de segmentation est utilisée pour valider le caractère. Dans le cas des lettres à trois primitives, la segmentation est automatique.

Lettre	Décomposition	Séquence
g		
h		

Tableau II-2. Décomposition des lettres 'g' et 'h' en primitives Glyph.

Pour avoir une analogie très fidèle à la forme et à l'ordre spatio-temporel de l'écriture tout en représentant un grand nombre de caractères, ces derniers ont été divisés selon leur nature en plusieurs catégories qui correspondent à des modes de saisie différents sur le clavier :

- des caractères alphanumériques,
- des caractères mathématiques,
- des caractères spéciaux,
- des caractères d'alphabet grec.

La technique de diviser les caractères en plusieurs modes est couramment utilisée. Beaucoup de méthodes existantes disposent de boutons supplémentaires pour le changement de mode (« abc » et « 123 » par exemple). Certaines méthodes à touches physiques utilisent des modes kinesthésiques (par exemple, avec le clavier AZERTY pour ordinateur portable, on peut appuyer simultanément sur la touche « Shift » et la touche

« & » pour saisir le chiffre '1'. Dans Glyph, c'est la touche de commande qui permet, entre autres, de changer de mode. La Figure II-4 montre la table principale des caractères alphanumériques Glyph. Les autres tables de caractères sont disponibles à l'annexe 2.

a	b	c	d	e	f	g	h	i	j
ā	b̄	c̄	d̄	ē	f̄	ḡ	h̄	ī	j̄
k	l	m	n	o	p	q	r	s	t
k̄	l̄	m̄	n̄	ō	p̄	q̄	r̄	s̄	t̄
u	v	w	x	y	z	1ère primitive 2ème primitive 3ème primitive			
ū	v̄	w̄	x̄	ȳ	z̄				
0	1	2	3	4	5	6	7	8	9
0̄	1̄	2̄	3̄	4̄	5̄	6̄	7̄	8̄	9̄

Figure II-4. Table des caractères alphabétiques et numériques Glyph.

Glyph est une méthode indépendante du dispositif. Elle peut être implémentée sous différentes formes et sur différents types de dispositifs. La Figure II-5 montre quelques implémentations imaginées par Uguen et Poirier [2004]. Seule une version logicielle pour PDA a été implémentée.

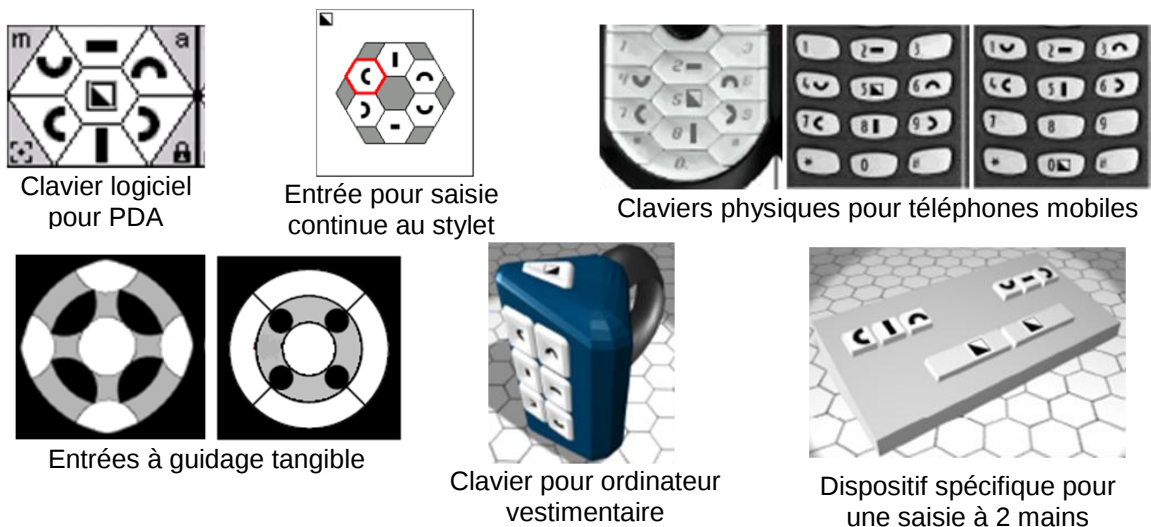


Figure II-5. Différentes implémentations possibles de Glyph.

4. Glyph2, l'amélioration

Avec Glyph, l'utilisateur doit effectuer trois opérations mentales concomitantes. La première est de décider du nombre de primitives qui codent un caractère, la deuxième est de choisir ces primitives et appuyer sur les touches correspondantes, et la troisième consiste à terminer la saisie en appuyant sur la touche de segmentation dans le cas où le caractère serait codé par une ou deux primitives. Cette dernière opération correspond à une tâche dite « articulatoire » [Coutaz, 1998], seulement nécessaire au système pour interpréter le flux des saisies, mais inutile pour la représentation du caractère par l'utilisateur. Inévitablement, ce type de tâches induit à de fréquentes erreurs.

L'autre inconvénient de Glyph est son KSPC élevé. Si on prend en considération les fréquences des lettres dans la langue française, la saisie avec un clavier Glyph non gestuel s'effectue en 2,75 KSPC contre 2,23 KSPC pour le clavier téléphonique multitap. Notre objectif avec Glyph2 était donc de réduire le nombre moyen d'appuis de touches par caractère, et, par conséquent, d'augmenter la vitesse de saisie. Nous avons réduit le nombre maximum de primitives par caractère à exactement deux primitives pour les caractères les plus fréquemment utilisés, à savoir, les lettres et les chiffres.

De toute évidence, le nombre de séquences d'une longueur de 2 qu'on peut obtenir à partir de six primitives est inférieur au nombre de séquences d'une longueur variable de un à trois. Avec des séquences d'exactly deux primitives, il est possible de coder au maximum 36 caractères (6^2), alors qu'avec des séquences variables de 1 à 3 primitives, on peut coder 258 caractères comme nous l'avons expliqué précédemment. Afin de maintenir une bonne analogie graphique et scripturale, la solution classique consiste à diviser les caractères de la table alphanumérique en deux tables distinctes (voir Figure II-6) :

- Une table des caractères alphabétiques qui contient les lettres en plus des caractères les plus fréquemment utilisés lors de la saisie de texte comme l'espace et l'apostrophe,
- Une table des caractères numériques pour les chiffres et les symboles utilisés dans des contextes numériques comme les signes '+' et '-'.

a	b	c	d	e	f	g	h	i	j
k	l	m	n	o	p	q	r	s	t
u	v	w	x	y	z	esp	'	- 1 ^{er} trait - 2 ^e trait	

0	1	2	3	4
5	6	7	8	9
+	-	/	*	=

Figure II-6. Le codage des caractères alphabétiques et numérique avec Glyph2.

Dans la Figure II-6, les analogies les moins fidèles aux caractères représentés sont mises en surbrillance pour attirer l'attention de l'utilisateur et faciliter leur mémorisation. Pour certains caractères, surtout parmi les chiffres, nous avons pu proposer deux décompositions pour assurer plus de flexibilité, éviter certaines erreurs et dans certains cas améliorer la vitesse de saisie (cas où l'utilisateur doit double-cliquer sur la même primitive pour saisir un caractère). D'autres méthodes, comme EdgeWrite [Wobbrock et al., 2003], offrent cette possibilité (voir annexe 1).

Avec le nouveau codage, chaque caractère est décomposé en exactement deux primitives. L'utilisateur n'a plus à se poser la question de combien de primitives il doit entrer et la segmentation est automatisée, mais un autre type de tâches articulatoires est apparu. En effet, pour les caractères qui correspondent à une seule primitive ('c', 'i', 'n', 'u' et l'espace), une deuxième est rajoutée pour atteindre la longueur normalisée des séquences. Cette deuxième primitive n'a aucune signification pour l'utilisateur vu

qu'elle ne rentre pas dans le principe de l'analogie. Elle est significative uniquement pour le système en permettant d'automatiser la segmentation.

Pour évaluer cette nouvelle version de Glyph, une expérience avec des utilisateurs novices a été menée pour comparer des implémentations de Glyph2 et de Glyph sur claviers non gestuels à touches avec le clavier téléphonique multitap [Poirier et Belatar 2006a]. Sur les trois sessions de l'expérience, les sujets ont été les plus rapides avec la méthode multitap avec environs 9,5 WPM sachant que cette méthode leur est familière. Avec Glyph2 la vitesse de saisie a atteint 7 WPM. Cette vitesse correspond à celle remarquée à la première session avec la méthode multitap. Cela indique une bonne mémorisation de l'alphabet Glyph2 et une facilité d'adoption de cette méthode. Avec Glyph la vitesse de saisie est restée à environ 4 WPM.

5. Glyph2 pour Pocket PC

Dans cette section, nous présentons Glyph2PPC, l'implémentation de Glyph2 sur Pocket PC. C'est une entrée gestuelle de type unistroke avec une interface à six touches de primitives et une touche de commandes. Pour la touche de commande, nous avons adopté le symbole '⌘' utilisé dans plusieurs autres méthodes comme T-Cube [Venolia et Neiberg, 1994] et le clavier des ordinateurs Mac. Les six touches de primitives sont organisées dans un espace hexagonal (voir Figure II-7) autour de la touche de commande pour optimiser, selon la loi de Fitts, tous les gestes reliant deux primitives et pour faciliter l'apprentissage (mémorisation visuelle). Les espaces inoccupés par l'hexagone dans la surface rectangulaire du clavier sont exploités pour assurer chacun une fonction spécifique :

- L'espace en haut à gauche assure un feedback sur le type de caractères saisis. Puisque l'interface du clavier ne change pas quand l'utilisateur bascule entre la saisie de lettres, de chiffres ou d'autres types de caractères. Ce feedback constitue un repère important pour l'utilisateur. À chaque type de caractères correspond une abréviation représentative :
 - « Abc » pour les caractères alphabétiques en commençant par une majuscule ;
 - « ABC » pour les caractères alphabétiques avec une saisie entièrement en majuscules ;
 - « abc » pour les caractères alphabétiques en minuscules ;
 - « 0-9 » pour les caractères numériques ;
 - « αβ » pour les caractères de l'alphabet grec ;
 - « ∑ » pour les caractères mathématiques ;
 - « @ » pour les caractères spéciaux.
- L'espace en bas à gauche permet d'ouvrir un menu déroulant pour d'accéder à des fonctionnalités supplémentaires d'information et de configuration. L'icône utilisée pour ce menu peut être facilement perçue comme « quelque chose qui va s'ouvrir » car elle symbolise généralement des actions de « tiroir » ;
- L'espace en bas à droite permet d'afficher la carte des caractères correspondante à la saisie active. C'est une aide facile d'accès pour permettre aux utilisateurs novices de consulter les cartes de caractères lors de la saisie ;

- L'espace en haut à droite affiche une icône qui permet de fermer le clavier.



Figure II-7. L'interface principale de Glyph2PPC.

Pour saisir du texte avec Glyph2PPC, l'utilisateur doit « sélectionner » les deux primitives correspondantes. Vu que la saisie gestuelle au stylet permet de valider une séquence de primitives au lever du stylet et que la primitive supplémentaire ajoutée à certains caractères dans Glyph2 n'était pas nécessaire, nous avons supprimé cette dernière. Les caractères correspondants à une seule primitive sont saisis en un simple clic et les autres sont saisis en un geste reliant les deux primitives correspondantes.

Pour adapter la méthode au niveau d'expérience de l'utilisateur et pour permettre d'atteindre de bonnes vitesses de saisie, trois types de saisie différents sont mis à disposition :

- Une saisie dite « statique » où l'utilisateur relie tout simplement les deux primitives (voir Figure II-8-a) ;
- Une saisie dite « dynamique » où le clavier se recentre par rapport au stylet au moment où l'utilisateur sélectionne la première primitive. Ce dernier n'a donc à faire qu'un simple geste du centre du clavier vers la deuxième primitive, cela permet de réduire l'amplitude du geste pour les primitives qui ne sont pas voisines sur le clavier. Tous les gestes sont donc de la même amplitude dans cette configuration. Le principe est similaire à celui des « pie-menus » (voir Figure II-8-b) ;
- Une saisie dite « dynamique de niveau expert » où l'utilisateur reproduit les mêmes gestes de la configuration dynamique mais sans que le clavier se repositionne. En effet, nous avons remarqué que le repositionnement du clavier devient perturbant quand l'utilisateur acquiert les automatismes du clavier dynamique (voir Figure II-8-c).

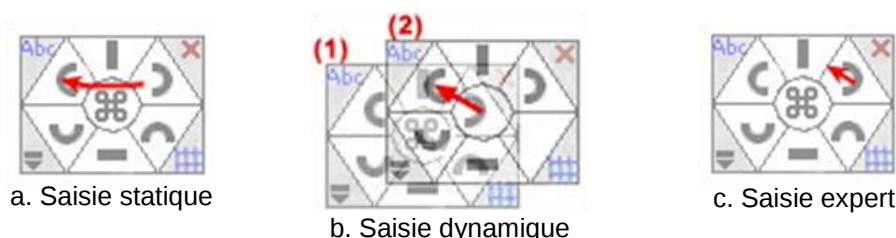


Figure II-8. Exemple de la saisie de la lettre 'x' avec les différents modes de saisie de Glyph2PPC.

Dans toutes les configurations, la saisie du caractère 'espace' correspond à un simple clic sur la touche de commande. Cette touche permet aussi d'explorer les différents modes de saisie (alphabétique, numériques...) et d'autres fonctionnalités communes dans

la saisie de texte. Au contact du stylet avec la touche de commande, le mode de commande est activé, un geste simple (↗) ou double (↖↗) permet d'accéder à une fonction ou un mode de saisie comme le détaille la Figure II-9.

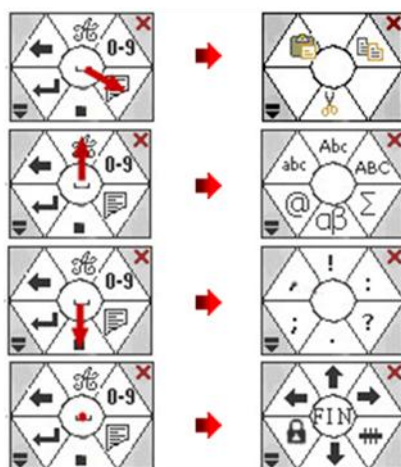


Figure II-9. Accès aux différents modes et fonctions via le mode de commande.

Glyph2PPC a été évaluée avec sept utilisateurs novices [Belatar et Poirier 2007a] sur une tâche de recopie en utilisant un jeu de 5 phrases simples (voir Tableau II-3). Les participants à cette évaluation sont âgés de 22 à 26 ans et ont un bon niveau de connaissance en informatique mais ils n'ont jamais utilisé un PDA avant cette expérience (stagiaires au VALORIA 2005). L'évaluation s'est déroulée en cinq sessions. Chaque session correspond à la saisie des cinq phrases de 124 caractères au total, espaces et retours à la ligne inclus (pour plus de détails sur l'évaluation et sur les résultats, voir [Belatar, 2005]).

Les résultats obtenus donnent une vitesse de saisie moyenne de 6,96 WPM à la cinquième session. La vitesse maximale atteinte par l'un des utilisateurs à la cinquième session est de 10,05 WPM. Le nombre moyen de gestes par caractère (GPC) obtenu à la cinquième session est de 1,08. Cela montre un très faible taux d'erreurs vu que le GPC caractéristique est théoriquement égale à 1. La majorité des erreurs constatées sont dues à la permutation de l'ordre d'entrée des deux primitives correspondantes à certains caractères. La saisie de la lettre 'd' par exemple représente 13,33% des erreurs constatées car la séquence de primitives correspondante (I + C) est souvent inversée, ce qui correspond à la saisie de la lettre 'a' (C + I).

Il a vu partir un incendie
Vous avez des yeux bleus
Un kayak est parti en mer
Que le meilleur gagne
Tous les chats sont gris

Tableau II-3. Les phrases utilisées pour l'évaluation du Clavier Glyph2PPC.

6. Conclusion

L'analogie aux formes des caractères est une technique qui a démontré son utilité de faciliter l'apprentissage dans plusieurs méthodes de saisie de texte. Sa combinaison avec une décomposition des caractères en primitives permet de construire des claviers réduits pour dispositifs mobiles. Plusieurs méthodes ont combiné ces deux techniques mais elles n'ont pas réussi à les exploiter d'une façon très optimale.

La méthode Glyph a exploité ces deux techniques pour offrir des claviers pour différents dispositifs. Sa version optimisée sur Pocket PC permet une saisie à un geste par caractère avec une vitesse de près de 7 WPM au niveau novice. Cette vitesse est comparable à celle observée avec d'autres méthodes [Martin et Pecci, 2007] mais reste trop faible pour remplacer des méthodes déjà préinstallées sur les dispositifs mobiles.

Les méthodes de la famille Glyph peuvent être implémentées sur différents dispositifs, mais à part l'implémentation gestuelle, elles présentent l'inconvénient d'un nombre d'appuis par caractère largement supérieur à 1 comme c'est le cas de la plupart des claviers réduits. Dans la suite de ce mémoire, nous présentons une nouvelle méthode de saisie qui tient en compte tous les pré-requis d'une méthode de saisie pour dispositifs mobiles en se basant sur les points forts et les points faibles des méthodes existantes et sur notre expérience avec Glyph.

CHAPITRE III

UNIGLYPH : UNE MÉTHODE DE SAISIE « OPTIMALE »

Résumé

Ce chapitre décrit les étapes de la conception d'une nouvelle méthode de saisie appelée « UniGlyph ». Il détaille ses bases, ses principes, son implémentation sur un dispositif numérique à écran tactile et son évaluation. Cette nouvelle méthode reprend un ensemble de principes déjà utilisés par d'autres méthodes, comme l'analogie, la décomposition des caractères en primitives et l'ambiguïté des touches pour créer une méthode singulière qui permet une saisie efficace en minimisant l'effort de l'utilisateur et l'espace sur le dispositif, tout en assurant un apprentissage facile et une saisie rapide.

1. Introduction

Comme on peut le constater dans les chapitres précédents, beaucoup d'alternatives sont apparues ces dernières années pour résoudre les problèmes de la saisie de texte sur dispositifs mobiles. Ces problèmes ont pour causes principales les dimensions réduites, les ressources limitées et les conditions non stables de l'utilisation de ces dispositifs.

Dans un premier temps, le clavier classique de l'ordinateur de bureau a été importé sur ces dispositifs mais ses grandes dimensions, le nombre important de ses touches ainsi que les petites dimensions de ces dernières font de lui un clavier très encombrant et peu pratique sur un dispositif de petite taille même avec les différentes améliorations qu'on a pu lui apporter. Ensuite, en profitant des interfaces tactiles sur certains dispositifs, des techniques de saisie basées sur la reconnaissance de caractères ou de formes ont été développées. Ces techniques sont généralement lentes et présentent des taux d'erreur importants dus à la variabilité de l'écriture et aux performances limitées des algorithmes de reconnaissance.

Les autres méthodes de saisie développées par la suite, spécialement pour ces dispositifs s'attaquent chacune à un ou plusieurs des problèmes posés sans réussir à satisfaire totalement les besoins de la saisie en mobilité. Certaines méthodes tentent de remédier au problème de la lenteur en proposant des organisations optimales des touches basées sur des données statistiques de la langue utilisée comme les claviers

MessageEase¹ et FITALY² par exemple. Le problème reste que ces claviers prennent autant ou même plus d'espace que le clavier logiciel traditionnel et que l'apprentissage n'est pas trivial car ces nouvelles distributions de lettres ne représentent aucune logique pour l'utilisateur. D'autres méthodes ont opté pour la minimisation des dimensions du clavier au prix d'un apprentissage plus difficile ou d'une saisie plus lente. Enfin, d'autres techniques accordent plus d'importance à la vitesse de saisie au prix de la difficulté d'apprentissage et des dimensions du clavier en proposant par exemple une saisie gestuelle de niveau mot.

L'indépendance du dispositif et la prise en compte des différents besoins des utilisateurs est aussi un critère très important pour l'adoption d'une méthode. Les auteurs d'EdgeWrite™ ont bien pris en compte ce critère tout en s'appuyant sur l'analogie aux caractères romains pour faciliter l'apprentissage sauf que sa table d'alphabet n'est pas vraiment optimale. Chaque caractère nécessite un nombre important de gestes ou d'appuis pour la saisie.

La méthode Glyph avec les différentes versions développées ces dernières années s'est basée aussi sur l'analogie aux caractères romains en les décomposant en primitives. Cette technique a permis de réduire l'effort mécanique fourni pour la saisie mais la vitesse reste insuffisante pour les tâches communément effectuées sur un appareil d'infocommunication mobile, à savoir : l'envoi de messages courts, de courriers électroniques ou la discussion instantanée entre autres.

Dans la suite, nous proposons une nouvelle méthode de saisie de texte sur dispositifs mobiles appelée UniGlyph où nous essayons de prendre en compte toutes les critiques faites précédemment sur Glyph. Dans un premier temps nous faisons un recueil et une sélection des principes et des techniques à retenir. Puis, nous décrivons une étude théorique et empirique de la faisabilité de cette méthode. Ensuite, nous décrivons son implémentation pour Pocket PC et son évaluation avant d'aborder les autres implémentations possibles.

2. Une méthode « optimale »

2.1. Règles de conception

Pour concevoir cette nouvelle méthode de saisie, nous avons commencé par fixer les buts à atteindre en se basant sur les carences soulignées dans l'état de l'art et par l'analyse de Glyph et Glyph2. La qualification d'« optimale » dans le titre de ce chapitre veut dire que la nouvelle méthode à concevoir assure le meilleur compromis entre les critères suivants :

- Une méthode pour dispositifs mobiles qui n'occupe pas beaucoup d'espace sur le dispositif dans le cas de solution physique ou sur l'écran du dispositif dans le cas d'une solution logicielle,

¹ ExIdeas Inc., page web de MessageEase : <http://www.exideas.com/ME/index.html>, visitée le 17/10/2008.

² Textware Solutions, page web : <http://www.fitaly.com/>, visitée le 17/10/2008.

- Une méthode indépendante du dispositif pouvant être implémentée sous différentes formes avec différentes modalités,
- Une méthode avec un KSPC ou un GPC minimal permettant de saisir du texte avec un minimum d'effort. L'idéal serait une saisie à 1 appui par caractère comme avec le clavier normal d'un ordinateur de bureau,
- Une méthode qui permet une saisie rapide adaptée aux applications d'un dispositif d'infocommunication mobile,
- Une méthode facile à apprendre pour éviter le rejet immédiat de l'utilisateur,
- Une méthode qui peut être utilisée par des personnes handicapées : l'objectif principal est d'atteindre les personnes avec des handicaps moteurs lourds vu que très peu de solutions ont été proposées à cette catégorie d'utilisateurs et que ces appareils informatiques mobiles peuvent leur être d'une très grande utilité. La nouvelle méthode doit être donc conçue de façon à être facilement adaptable au handicap.

Pour répondre à ces critères, nous avons étudié les techniques utilisées par les méthodes de saisie pour dispositifs mobiles et celles utilisées par les solutions proposées aux personnes handicapées. Cette étude nous a permis de choisir les principes de la nouvelle méthode de saisie. La Figure III-1 détaille ces choix.

Dans un premier temps nous retiendrons seulement les techniques utilisées dans la saisie de texte sur dispositifs mobiles pour personnes valides. Les autres techniques (concernant la saisie adaptée aux personnes handicapées) seront adjointes et détaillées dans le chapitre suivant. Les techniques retenues pour la nouvelle méthode sont donc :

- Utilisation d'un principe d'analogie avec une décomposition des caractères en primitives,
- Une saisie à désambiguïsation lexicale,
- Un nombre de touches réduit au minimum possible,
- Exploitation du gestuel,
- Utilisation de données linguistiques,
- Utilisation de feedbacks visuels et sonores,
- Une entrée de niveau mot,
- Conformité à la loi de Fitts.

2.2. Mise en œuvre

Pour mettre en œuvre cette sélection de techniques et atteindre tous les objectifs fixés pour la nouvelle méthode de saisie, notre idée de base consiste à décomposer les caractères en formes de base (primitives) selon un principe d'analogie aux caractères latins, puis choisir une seule primitive pour représenter chaque caractère. Puisque le nombre de ces primitives ne peut pas être inférieur à six (comme dans Glyph et les autres méthodes existantes basées sur la décomposition analogique), on peut les regrouper sur un nombre minimal de touches et utiliser un système de désambiguïsation pour résoudre l'ambiguïté des touches comme dans toutes les méthodes de ce type (ex. T9).

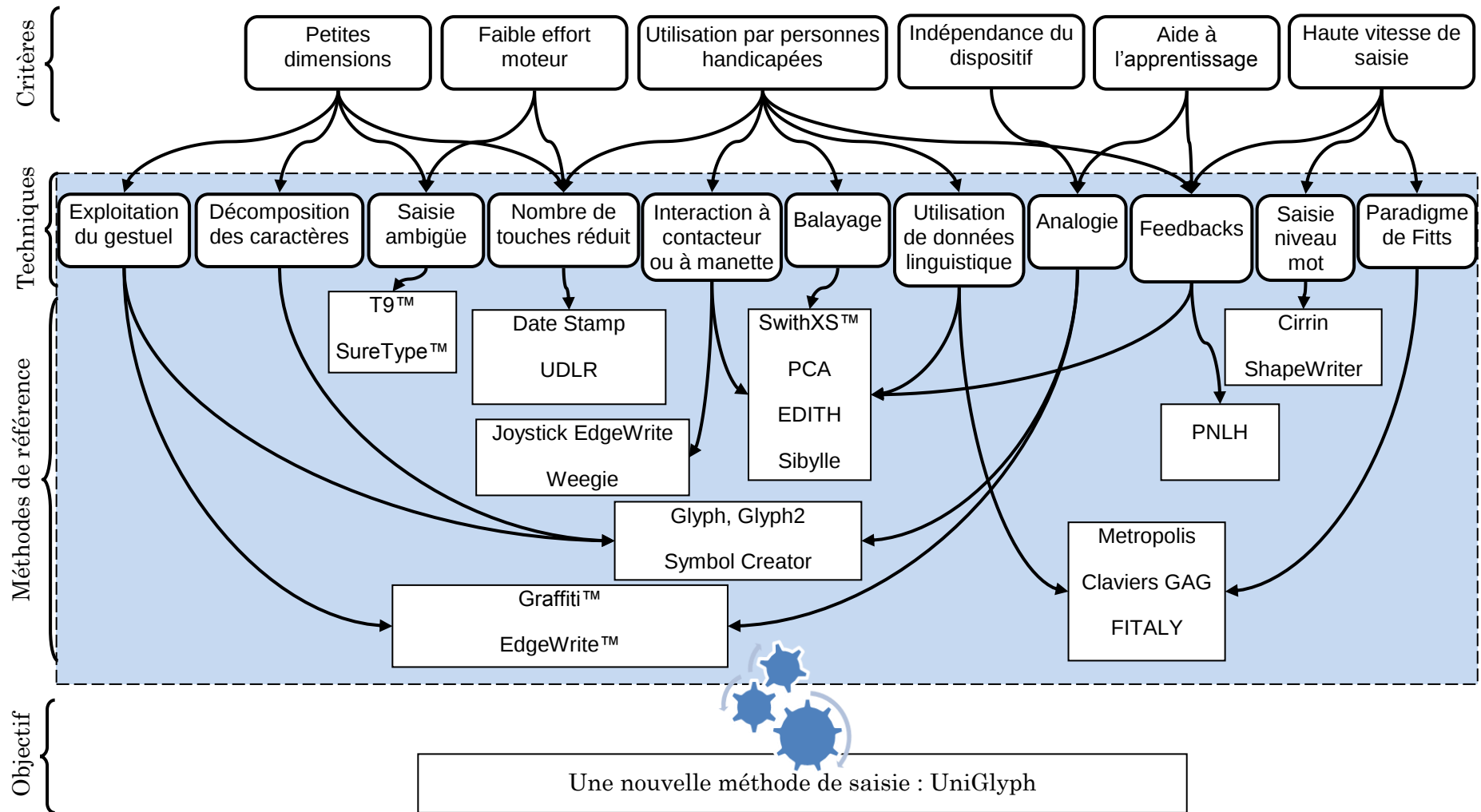


Figure III-1. Schéma du principe et des bases de la conception d'UniGlyph.

Un système de prédiction linguistique peut renforcer le système de désambiguïsation pour que les mots les plus probables soient les plus facilement accessibles. Ensuite, on peut implémenter des commandes gestuelles pour réduire le KSPC sans augmenter le nombre de touches et enfin, il faudra implémenter un mécanisme d’affichage adapté pour proposer la liste des mots les plus probables à l’utilisateur.

La nouvelle méthode est donc une méthode à touches ambiguës avec une saisie de niveau mot. Plusieurs autres méthodes utilisent ces techniques mais la différence d’UniGlyph avec ces méthodes est que :

- Avec UniGlyph, le nombre de labels par touche sera très limité : chaque touché contiendra seulement deux primitives. Avec les autres méthodes ambiguës, plus le nombre de touches est réduit plus le nombre de labels sur les touches est élevé. Par exemple, avec le clavier SureType™ du Blackberry™, chacune des 14 touches ambiguës contient entre 1 et 2 lettres ; avec le clavier T9, chacune des 8 touches ambiguës contient 3 à 4 lettres et avec GORSUV [Dunlop, 2004], chacune des 4 touches ambiguës est libellée de 6 à 7 lettres. Chaque touche peut bien sûr correspondre à un nombre plus important de caractères dans le cas des langues qui utilisent beaucoup de lettres accentuées (par exemple avec le T9 en français, la touche ‘2’ correspond aux caractères : a b c ä à æ ç...) mais généralement, ces caractères ne sont pas affichés sur les touches.
- Pour la majorité des méthodes ambiguës à très peu de touches, les concepteurs se trouvent obligés d’assigner les lettres aux touches de façon à optimiser les résultats de la désambiguïsation (en utilisant différents algorithmes d’optimisation). Ces organisations ne sont ni familières ni logiques pour l’utilisateur. Le temps de la recherche visuelle devient non négligeable dans ce type de claviers, ceci peut entraîner un rejet rapide de la méthode. Avec UniGlyph, chaque touche est libellée de deux primitives seulement ce qui rend le temps de la recherche visuelle minimal, même pour un utilisateur novice.

Cependant, avec UniGlyph, l’étiquetage des touches par des primitives présente l’inconvénient pour l’utilisateur de ne pas pouvoir s’appuyer directement sur sa « connaissance du monde » [Norman, 1980] pour trouver la bonne touche par la simple lecture des labels sur les touches du clavier. Il doit avoir en tête le principe de décomposition en primitives et choisir la bonne primitive pour entrer une lettre. Il doit donc s’appuyer sur ses « connaissances en tête », qu’il a acquises par apprentissage.

À part les méthodes de saisie directe sur clavier complet à un caractère par touche, toutes les autres méthodes de saisie présentent plus ou moins ce même inconvénient. Le postulat adopté ici est qu’il est plus avantageux dans un contexte d’interaction dégradée de disposer d’une interface de saisie très limitée plutôt que d’utiliser un clavier plus ou moins complet qui ne nécessiterait aucun apprentissage. Il est à noter que le fait de libeller des touches ambiguës directement par des lettres ne signifie pas que la méthode ne nécessite pas d’apprentissage.

Il reste maintenant à savoir quelle analogie adopter et comment choisir la primitive qui va représenter chaque lettre de façon à faciliter cet apprentissage. Une première idée

consiste à reprendre les mêmes décompositions utilisées par Glyph ou l'une de ses versions améliorées, choisir une seule primitive pour chaque caractère, puis les combiner en deux par touche pour avoir 3 touches de primitives au total (puisque Glyph utilise 6 primitives de forme). Cette idée a été vite écartée quand on s'est aperçu que le choix de la bonne primitive, parmi les trois (au maximum) qui constituent chaque caractère avec Glyph ou parmi les deux primitives qui constituent chaque caractère avec Glyph2, ne serait pas évident pour l'utilisateur. En effet le choix des primitives doit respecter deux critères :

- Une désambiguïsation efficace : les combinaisons de primitives doivent être choisies de façon à assurer un résultat de désambiguïsation optimal. Autrement dit, le nombre de mots présentés à l'utilisateur après la saisie d'une séquence de primitives doit être minimal en moyenne,
- Une mémorisation facile : le choix des primitives doit être « logique » pour l'utilisateur afin qu'il puisse s'en souvenir sans difficultés. Pour choisir un seul trait dans le dessin d'un caractère, l'utilisateur doit disposer d'une règle simple ou d'une logique claire pour ne pas hésiter entre les différents traits qui composent le dessin du caractère.

Pour répondre à ces deux critères nous avons procédé à deux études complémentaires pour fixer le jeu de primitives de la nouvelle méthode. Une première étude « théorique » basée sur des données statistiques de la langue pour étudier l'efficacité de la désambiguïsation et une seconde étude expérimentale avec des utilisateurs pour étudier la facilité de mémorisation du jeu de primitives.

3. Études au préalable

3.1. Étude statistique

Dans cette étude statistique, la première étape était de définir plusieurs jeux de primitives résultant de différentes analogies aux lettres en minuscules ou en majuscules avec des logiques de décomposition différentes, telles que :

- Choisir la première primitive dans l'ordre spatial de l'écriture (par exemple la lettre 'M' est représenté par une primitive verticale 'l').
- Choisir la primitive considérée comme la plus prégnante (caractéristique) de la forme de chaque lettre (par exemple la lettre 'M' est codée par une primitive droite inclinée '\ ' ou '/').
- Choisir les traits arrondis d'abord : si la lettre contient une boucle ou un trait arrondi, elle est représenté par une des primitives suivantes : « C, D, O, U » (par exemple la lettre 'B' est codée par la primitive 'D'). Si la forme de la lettre ne contient aucune courbe (lettre 'T' par exemple), elle est codée par une primitive représentant l'absence de boucles ('-' ou 'l' per exemple).
- Choisir le moins primitive commune dans l'alphabet de codage. Par exemple, la lettre «M» est représenté par la primitive '\ ' ou '/ ' parce que ces primitives sont moins utilisées que la primitive 'l' dans le codage des lettres (les primitives '\ ' et '/ ' sont présentes dans 10 / 26 lettres contre 17/26 pour '-' et 'l').

Le Tableau III-1 montre quelques exemples des solutions avec 3 à 5 touches qui illustrent les logiques de décompositions expliquées ci-dessus. La colonne de droite exprime le résultat donné par chaque solution en termes de nombre de mots possibles par séquence de clics sur un dictionnaire de français de 250 000 entrées. Les autres résultats statistiques obtenus sont détaillés dans l'annexe 3.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	
 1 ^{re} primitive	∩		∩		∩	∩						∩	∩	∩			∩	∩		∩	∩	∩	∩	∩	∩	∩	2,12
 Plus prégnant	∩	∩	∩	∩	∩		∩	∩					∩	∩	∩	∩	∩		∩		∩	∩	∩	∩	∩	-	2,36
 boucles	∩	∩	∩	∩	∩	∩	∩	∩		∩	∩	∩	∩	∩	∩	∩	∩		∩		∩	∩	∩	∩	∩	∩	1,90
 boucles	∩	∩	∩	∩	∩	∩	∩	∩		∩	∩	∩	∩	∩	∩	∩	∩		∩		∩	∩	∩	∩	∩	∩	3,87
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
 priorité	/	∩	∩	∩			∩				/		∩	∩	∩	∩	∩	∩	∩	-		∩	∩	∩	∩	/	2,52
 priorité	∩		∩		-	-	∩				∩	∩	∩	∩	∩	∩	∩	∩	∩	-		∩	∩	∩	∩	-	1,51

Tableau III-1. Exemples de jeux de primitives et de touches étudiés.

Après l'étude statistique, nous avons retenu les deux tables de caractères de la Figure III-2 et la Figure III-3 qui donnent un bon compromis entre nos trois critères de sélection, à savoir : la « logique » simple du choix de primitives pour faciliter la mémorisation, les résultats de la désambiguïsation en termes de nombre de candidats (mots) par séquence d'appuis et le nombre de touches utilisées.

La première solution (Figure III-2) est basée sur une analogie aux lettres manuscrites en minuscules avec un choix des boucles en premier. Elle donne une moyenne de candidats par séquence d'appuis de 3,87 avec un écart type de 11,49 sur un dictionnaire d'environ 250 000 mots en français¹. La deuxième (Figure III-3) est basée sur une analogie aux caractères imprimés en majuscules avec un ordre de priorité des traits que contient le dessin du caractère : on choisit les inclinés tout d'abord, sinon les arrondis et enfin les traits droits. Cette solution donne une moyenne de candidats par séquence d'appuis de 2,52 avec un écart type de 4,62 sur le même dictionnaire. Dans les figures suivantes (Figure III-2 et Figure III-3), les primitives qui représentent chaque lettre sont dessinées en couleur différente.

¹ ABU (Association des Bibliophiles Universels) dictionnaire d'environ 251 000 entrées en français. Disponible à : <http://abu.cnam.fr/>. Req. Nous avons effectué un prétraitement sur le dictionnaire original de façon à supprimer les entrées qui contiennent des caractères non alphabétiques ainsi que les mots composés.



Figure III-2. Analogie aux caractères manuscrits en minuscules.



Figure III-3. Analogie aux caractères imprimés en majuscules.

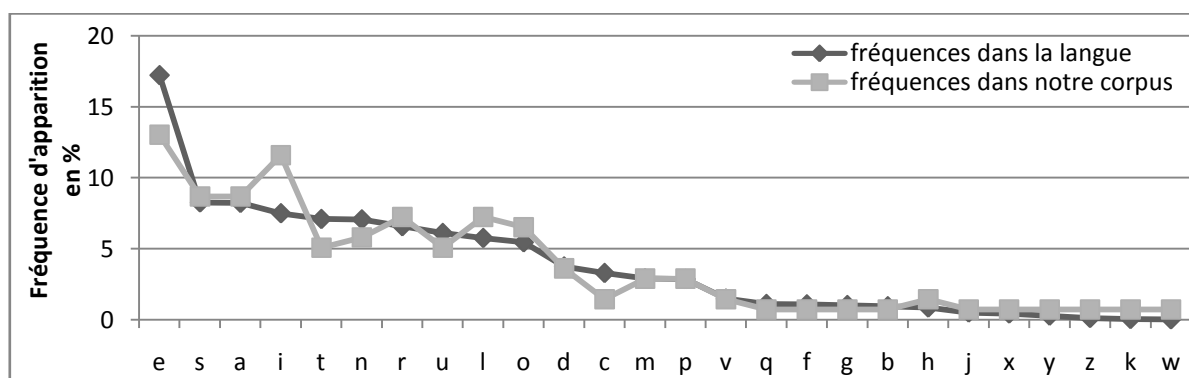
3.2. Évaluation expérimentale

Pour s'assurer que ces systèmes de codage sont faciles à mémoriser par les utilisateurs potentiels et en choisir le meilleur, nous avons mené une étude expérimentale en ligne avec des internautes. Dans la suite nous appelons « analogie minuscule » la solution illustrée par la Figure III-2 et « analogie majuscule » la solution illustrée par la Figure III-3.

Deux applications Web (Applets Java) ont été mises en place. Chacune permet aux utilisateurs de recopier un ensemble de 22 mots simples en Français au moyen de trois touches de primitives ambiguës (deux primitives par touche) et une touche de correction (suppression d'un caractère). Les 22 mots totalisent 138 caractères à saisir (Tableau III-2). Ils ont été choisis de manière à ce que les fréquences des lettres soient les plus conformes possibles aux fréquences des lettres observées dans la langue française¹ et qu'ils ne représentent aucune ambiguïté ou difficulté orthographique (voir Figure III-4).

bataille	trois	joli	dessus	inconnu	rappeler	petit	question	hasard	second	heureux
maladie	kiwi	alors	assez	dernier	remplir	ainsi	voyage	sommeil	violent	fauteuil

Tableau III-2. Liste des mots utilisés dans l'évaluation préliminaire en ligne.



¹ Nous nous sommes basés sur les fréquences des lettres mises à disposition sur le site www.lexique.org en additionnant la fréquence de chaque lettre avec celles de toutes ses formes accentuées (e.g. les fréquences des lettres 'e', 'é', 'è', 'ê' et 'ë' sont toutes additionnées pour donner la fréquence de la lettre 'e') car dans le principe d'UniGlyph toutes les formes accentuées sont saisies de la même manière que la lettre de base. Cet aspect sera détaillé dans la suite du mémoire.

Figure III-4. Fréquences des lettres dans les 22 mots de notre corpus par rapport aux fréquences des lettres dans la langue française.

Un appel à participation à cette expérience a été publié sur différents forums de discussion généralistes. Les internautes ont été invités à « tester » une nouvelle méthode de saisie. Nous avons choisi de faire une évaluation en ligne pour avoir un nombre maximal de participants de différents âges, de centres d'intérêts différents, qui utilisent facilement la souris. Cet environnement ne peut pas être utilisé pour évaluer une méthode de saisie vu le contrôle réduit des conditions de l'expérience mais dans ce cas, nous n'évaluons pas la méthode de saisie, nous effectuons seulement une étude préalable sur la facilité d'apprentissage des deux alphabets proposés.

Au premier accès d'un internaute sur la page web dédiée à l'expérience, il est invité à entrer un pseudonyme en plus de quelques informations générales comme :

- L'âge :
 - moins de 26 ans,
 - entre 26 et 40 ans,
 - plus que 40 ans.
- L'expérience avec le clavier AZERTY de l'ordinateur :
 - novice,
 - intermédiaire,
 - expert.
- L'utilisation ou pas d'un clavier virtuel. Si oui, lequel?
- L'utilisation de téléphone mobile. Si oui, combien envoie-t-il de SMS :
 - moins de 4 SMS par semaine,
 - entre 4 et 28 SMS par semaine,
 - plus de 4 SMS par jour.

À la fin du formulaire, l'internaute est invité à recopier une courte phrase qui nous permet de vérifier sa vitesse de saisie avec le clavier de son ordinateur. En validant le formulaire, le participant est enregistré dans notre base et l'une des deux solutions (analogie majuscule ou analogie minuscule) lui est attribuée automatiquement de façon aléatoire. Les participants ne sont pas informés de l'existence de deux solutions différentes. Lors des connexions ultérieures, l'accès d'un participant est contrôlé par pseudonyme et adresses IP pour le diriger sur la même solution qu'il a utilisée la première fois sans l'informer de l'existence d'une deuxième.

Après l'identification du participant, l'applet affiche l'alphabet correspondant en plus d'une brève explication du principe de la saisie de la méthode (Figure III-5 et Figure III-6, écran①).

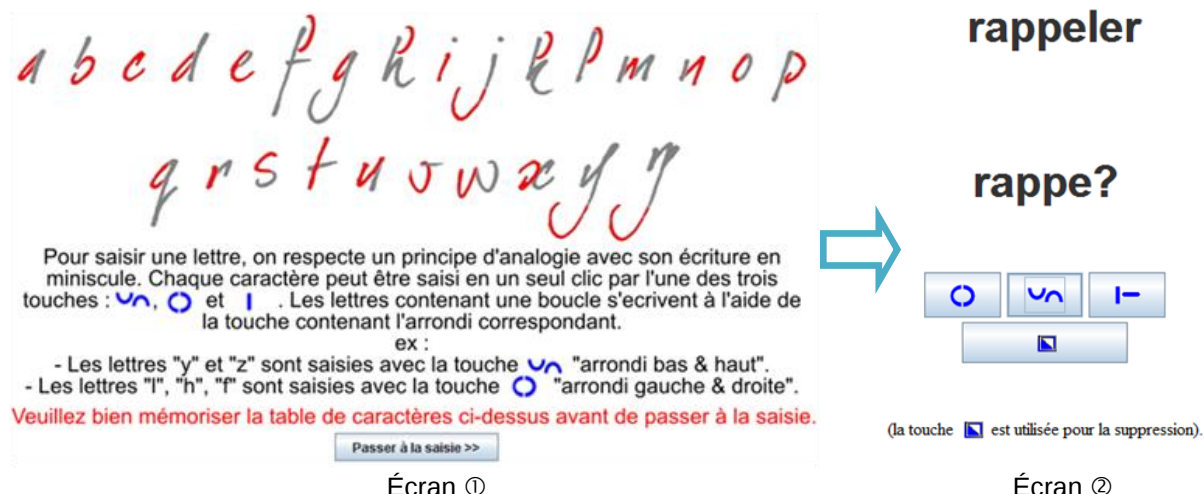


Figure III-5. Applet d'évaluation de l'analogie minuscule. Exemple de la saisie du mot « rappeler » avec une erreur sur la lettre 'l'.

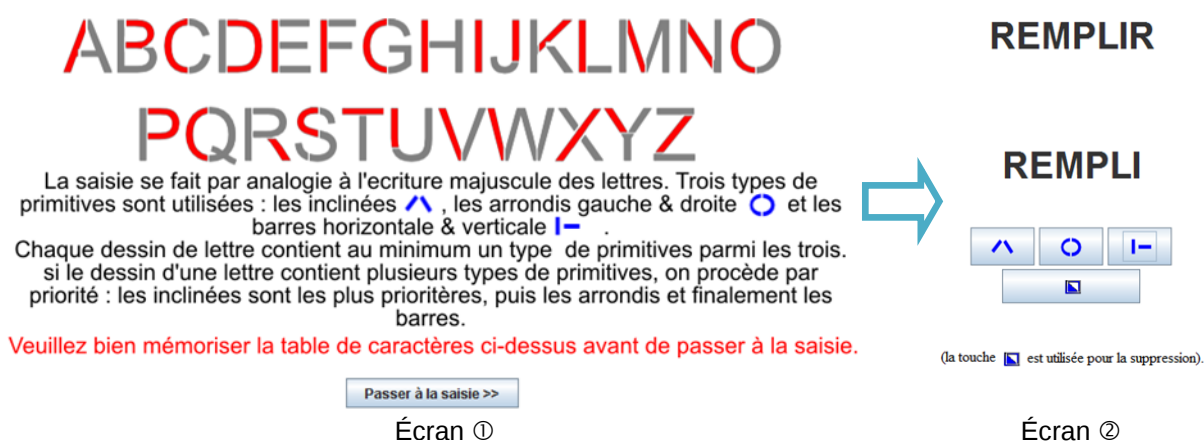


Figure III-6. Applet d'évaluation de l'analogie majuscule. Exemple de la saisie du mot « remplir ».

La page qui présente l'alphabet reste affichée deux minutes au maximum avant de passer à la saisie. Le participant peut aussi à tout moment interrompre cet écran et passer à la saisie en cliquant sur un bouton dédié. L'écran suivant lui permet de recopier les 22 mots proposés avec les quatre touches de la solution testée. Les mots sont proposés dans un ordre aléatoire à chaque session pour éviter leur mémorisation et ainsi l'anticipation de l'utilisateur. Aucune action de validation n'est demandée pour ne pas compliquer la tâche et pour assurer une fluidité de la saisie. À la fin de la saisie sans erreurs d'un mot, le mot suivant est automatiquement affiché. L'application n'utilise aucun système de désambiguïsation. L'utilisateur ne peut saisir que les mots proposés. Quand la touche appuyée ne correspond pas à la prochaine lettre à saisir, un point d'interrogation est affiché pour indiquer l'erreur, l'utilisateur doit la supprimer à l'aide de la quatrième touche pour continuer la saisie. Toutes les actions sont automatiquement enregistrées dans des fichiers de journalisation sur le serveur.

L'expérience a duré un mois, 117 internautes se sont inscrits, 58 parmi eux ont terminé au moins une session. Le Tableau III-3 donne une vue d'ensemble sur les profils de ces participants :

Information	Variantes	Pourcentage
Âge	Moins de 26 ans	70,07%
	Entre 26 et 40 ans	24,80%
	Plus de 40 ans	05,13%
Niveau d'expérience avec le clavier d'ordinateur	Novice	06,84%
	Intermédiaire	60,68%
	Expert	32,48%
Utilisation de claviers virtuels	Oui	02,56%
	Non	97,44%
Nombre de SMS envoyés	Non	16,24%
	Moins de 4 par semaine	47,86%
	Entre 4 par semaine et 4 par jour	30,77%
	Plus de 4 par jour	05,13%

Tableau III-3. Résumé des profils des participants.

Sur les 58 participants qui ont achevé au moins une session, 22 ont testé l'analogie minuscule et 36 l'analogie majuscule comme le montre le tableau ci-dessous :

	Sessions							Total
	1	2	3	4	5	6	7	
Applet de l'analogie minuscule	22	9	6	5	5	2	2	51
Applet de l'analogie majuscule	36	14	7	5	3	2	1	68
Nombre total de participants par session	58	23	13	10	8	4	3	119

Tableau III-4. Répartition des participations par applet et par session.

Le fait qu'il n'y ait que 58 participants sur 117 internautes enregistrés ne signifie pas forcément qu'ils ont eu des difficultés à utiliser cette méthode. En effet, dans le contexte d'une évaluation en ligne, il est normal que tous les participants ne soient pas vraiment motivés par l'expérience. La plupart d'entre eux vont sur le site uniquement par curiosité. Certains d'entre eux vont plus ou moins loin dans l'expérience, mais d'autres abandonnent dès qu'ils satisfont leur curiosité. Au total, 119 sessions exploitables (complètes) ont été enregistrées et 16 422 caractères ont été saisis.

Étant donné que l'objectif de notre expérience est d'évaluer la faisabilité de la nouvelle méthode et de choisir la meilleure solution entre les deux proposées, l'élément de performance le plus important à étudier est le taux d'erreurs. Le protocole de l'expérience a été fait pour atteindre cet objectif. Il oblige l'utilisateur, contrairement à une saisie « naturelle », à corriger toutes les erreurs lors de la saisie. Le taux d'erreurs résidentielles (MSD error rate) est donc de 0%. Cela nous permet d'étudier en détail la difficulté de saisie de chaque lettre avec chacune des deux solutions en calculant le « taux d'insertions corrigées » de chaque lettre ou Corrected Insertion Rate (CIR). Cette métrique a été introduite par Wobbrock et Myers [2006a] dans une étude qui a porté spécialement sur les taux d'erreurs de niveau caractère. Le CIR d'une lettre i donnée est

obtenu en divisant le nombre de corrections de cette lettre sur le nombre de ses apparitions dans le texte saisi :

$$CIR_i = \frac{Corrections_i}{Occurrences_i} \times 100\% \quad (1)$$

La Figure III-7 montre le taux de corrections effectuées sur chaque lettre. Ces dernières sont triées par fréquence dans la langue française. Les liaisons pleines indiquent que l'analogie majuscule donne de meilleurs résultats que l'analogie minuscule sur la lettre concernée et vice-versa pour les liaisons creuses.

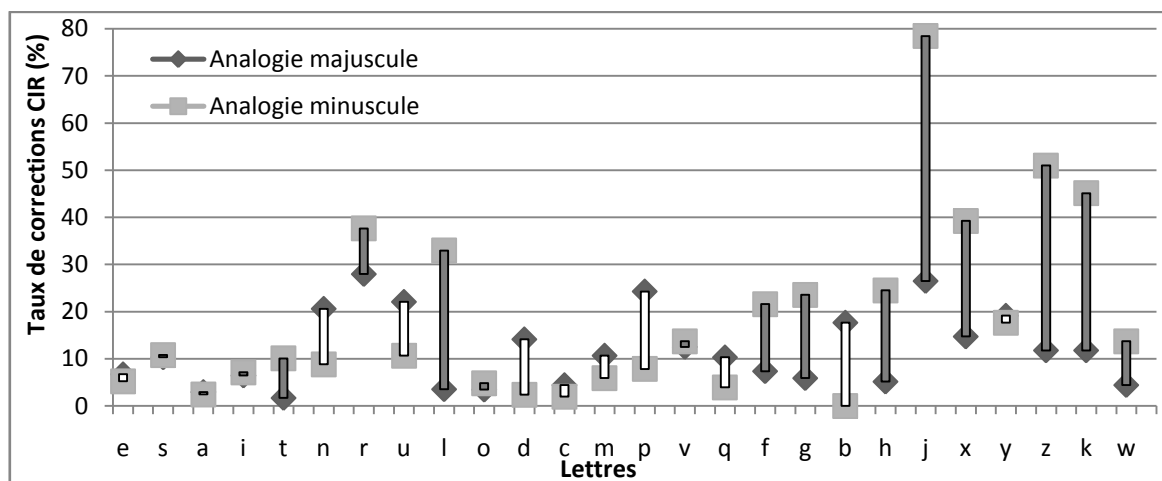


Figure III-7. Comparaison des taux de corrections par lettre avec les deux solutions proposées.

On peut remarquer que pour les lettres les plus fréquentes (e, s, a, ..., u), la différence entre les deux analogies n'est pas très significative. Les participants ont plus de difficultés à mémoriser les lettres les moins fréquentes surtout avec l'analogie minuscule. Seule la lettre 'r' représente un réel problème dans les deux analogies. Pour entrer 'r' selon l'analogie minuscule, les participants cliquent souvent sur la touche contenant la primitive « $\cap$ » au lieu de la primitive « $\mid$ » car ils considèrent que la lettre 'r' contient un trait arrondi. En revanche, pour la lettre 'l', ils appuient sur la primitive « $\mid$ » au lieu de l'arrondi « $\cap$ » qui représente la boucle de la lettre 'l'. Pour l'analogie majuscule, les participants ont parfois préféré la primitive associée au début de l'écriture, comme dans le cas de la lettre 'R', ils cliquent souvent sur les touches « $\neg$ » ou « $\cap$ » au lieu de la touche « $\vee$ ». La même erreur se répète avec la lettre 'P' que les participants associent à la touche « $\neg$ » au lieu de « $\cap$ ». En général, l'analogie majuscule semble être plus facile à mémoriser par les utilisateurs que ce soit pour les lettres les plus fréquentes ou les moins fréquentes.

La comparaison des taux d'erreurs globaux par session va aussi dans la même direction en affichant de meilleurs résultats avec l'analogie majuscule (voir Figure III-8). Ces taux d'erreurs sont calculés en divisant le nombre de corrections sur le nombre total de caractères saisis par tous les utilisateurs lors d'une session.

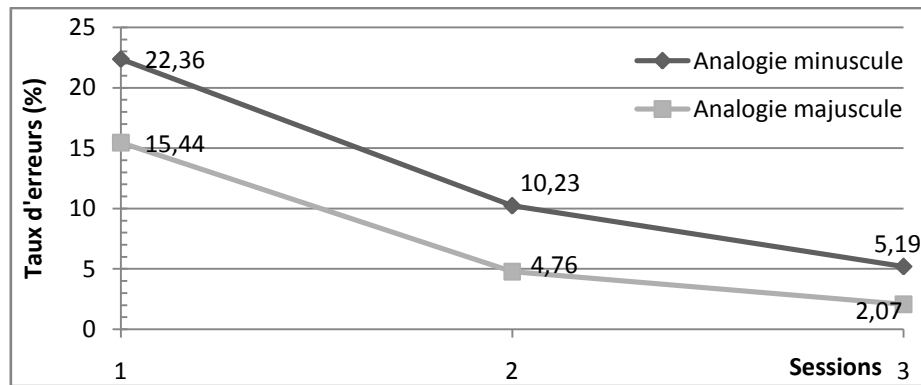


Figure III-8. Taux d'erreurs sur les trois premières sessions.

À la troisième session le taux d'erreur observé avec l'analogie minuscule 5,19% reste plus de deux fois plus élevé que celui observé avec l'analogie majuscule 2,07%. Nous ne considérons ici que les trois premières sessions car le nombre de participants n'est pas significatif au-delà. Les courbes gardent cependant la même tendance jusqu'à la septième session (2,17% avec l'analogie minuscule contre seulement 0,72% avec l'analogie majuscule). En plus des résultats empiriques, les commentaires laissés par les utilisateurs concernant l'analogie majuscule dégagent plus d'enthousiasme comme le montre le Tableau III-5.

Analogie majuscule	Analogie minuscule
- Après quelques mots, ça vient tout seul. Ha j'ai mieu celui la ☺ - C'est génial comme projet ! - J'adore le principe ! - Je viens d'essayer ; et j'avoue que c'est très intuitif et finalement très pratique. - J'ai fait un test je trouve sa bien ! Ce serait super pratique sur un PDA ! - Non sérieusement, ça paraît tout con mais c'est une idée qui a due prendre pas mal de temps a trouver	- Pourquoi le L ne ce fait-il pas avec le 3^{ème} boutons ? c'est assez deroutant.. ☺ - Ça devient vite lassant quand même, j'arrete ☺ - J'aime le concept ; mais par contre le L je m'y habitue pas je l'aurait mis en et pas en sur le bouton de gauche et inversement pour le r. - J'ai vraiment du mal avec les l et les r ☺

Tableau III-5. Les commentaires des participants (authentiquement recopiés).

Cette expérience nous permet aussi de comparer les deux solutions en termes de vitesse de saisie qui a atteint 10,35 WPM et 10,03 respectivement avec l'analogie majuscule et l'analogie minuscule à la troisième session mais cette donnée n'est pas très significative car ni l'interface ni le dispositif d'interaction n'est approprié. En outre, le protocole adopté (évaluation en ligne) ne permet pas un contrôle total des conditions de l'expérience et ainsi, il ne permet pas d'obtenir une vitesse de saisie empirique significative.

En se basant sur cette étude, nous avons décidé de retenir l'analogie majuscule pour l'alphabet UniGlyph. Le seul problème de cette solution est la lettre 'R' que les utilisateurs ont plus tendance à saisir à l'aide de la primitive arrondie. Nous avons

décidé de ne pas changer son codage car les utilisateurs le retiennent dès le troisième essai. Garder une règle claire de décomposition est aussi un facteur très important pour la mémorisation. La règle de codage pour l'analogie majuscule s'exprime ainsi :

- Si la lettre en majuscule contient un trait incliné alors appuyer sur la touche des « traits inclinés »,
- Sinon, si elle contient une boucle alors appuyer sur la touche des « traits arrondis »,
- Sinon appuyer sur la touche des « traits droits ».

4. Le clavier UniGlyph sur PDA

La conception de la méthode UniGlyph repose sur des critères généraux (décrits dans les sections précédentes) pour assurer une universalité et une indépendance du dispositif. Cela nous permet de cibler l'implémentation pour bien exploiter les possibilités qu'offre chaque dispositif et assurer un maximum de performance tout en utilisant les mêmes principes de base et la même règle de saisie. Les principales caractéristiques qu'offre un dispositif de type « assistant numérique » (PDA) qu'on peut prendre en compte dans notre contexte sont :

- Un écran qui prend la plus grande partie de la surface du dispositif pour permettre d'afficher tant bien que mal des documents, des pages web et d'autres applications qui nécessitent une visualisation large,
- Une possibilité d'interaction au doigt ou au stylet grâce à l'écran tactile. L'interaction au doigt est de plus en plus recherchée. Les nouveaux dispositifs de la catégorie du PDA (principalement l'iPhone) essaient de favoriser ce type d'interaction (full tactile) en abandonnant complètement le stylet,
- Quelques touches physiques représentées essentiellement par le pavé directionnel en bas de l'écran dans la grande majorité des PDA commercialisés actuellement.

Dans la suite nous décrivons une implémentation d'UniGlyph sur PDA qui prend en considération ces caractéristiques.

4.1. Description

4.1.1. L'interface et l'utilisation du clavier

4.1.1.1. Les éléments de l'interface

Le clavier virtuel UniGlyph pour PDA est composé de trois touches de primitives, une touche de commande et une zone d'affichage et de sélection des mots proposés que nous appellerons dans la suite « ZAS » (voir Figure III-9).

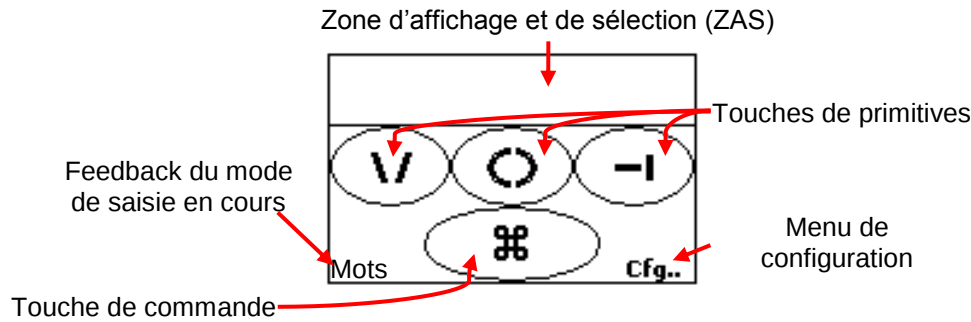


Figure III-9. Interface principale du clavier UniGlyph sur PDA.

Les dimensions des touches de primitives sont de l'ordre de 100 mm² pour permettre une saisie éventuelle au doigt et une bonne vitesse d'interaction selon la loi de Fitts. Cet élément est traité plus en détail dans la section 4.2.1. La touche de commande est de plus grande taille pour optimiser le trajet du doigt ou du stylet entre elle et les autres touches. La forme des touches en ellipses est virtuelle. La surface réellement sensible de chaque touche est rectangulaire. La forme ovale réduit visuellement la taille réelle des touches pour éviter les erreurs dues aux appuis entre touches dans le cas d'une saisie au doigt.

Les touches de primitives sont libellées de deux primitives chacune et organisées de gauche à droite selon l'ordre de priorité de ces dernières pour faciliter la mémorisation et l'utilisation de la règle unique de saisie présentée en fin de la section précédente.

La touche de commande permet de passer d'un mode à l'autre (voir les modes plus loin) et d'accéder à quelques fonctionnalités de base de saisie de texte, comme la suppression, l'espacement et le retour à la ligne. Le symbole de la touche de commande a été repris de Glyph2PPC qui l'avait déjà repris d'autres méthodes comme T-Cube [Venolia et Neiberg, 1994] ou le clavier du Macintosh (voir Figure III-10).

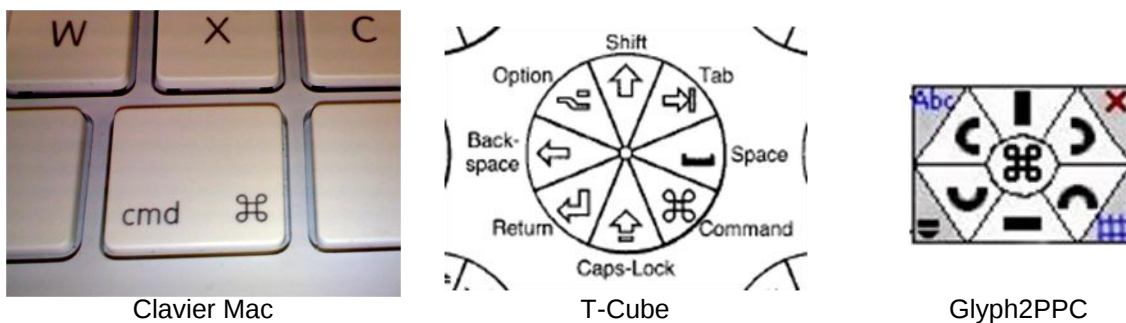


Figure III-10. Le symbole de la touche de commande sur différentes méthodes de saisie.

La ZAS quant-à-elle, permet d'afficher la liste des mots résultants de la désambiguïsation des séquences d'appuis effectués par l'utilisateur. Elle repose sur une technique de zoom de type « FishEye » [Sarkar et Brown, 1993]. C'est un affichage de type *focus+contexte* [Lamping et al., 1995] (aussi appelé *aperçu+contexte* [Hornbaek, et Frokjaer, 2001]) qui permet de visualiser 2 ou 3 mots de la liste tout en donnant un feedback sur l'existence ou non d'autres mots. Cette technique a été étudiée et a prouvé

son utilité dans différents contextes comme la navigation dans les graphes et les réseaux [Schaffer et al., 1996] ou l'utilisation de menus [Bederson, 2000]. Cependant, à notre connaissance, elle n'a jamais été utilisée dans une application de visualisation de listes de mots prédits par un outil de saisie de texte. Dans le clavier UniGlyph, nous avons décidé d'adopter cette technique pour bénéficier de sa capacité à afficher un grand nombre de données sur espace réduit. Elle nous permet aussi d'éviter des actions inutiles de l'utilisateur par exemple quand la fin de la liste est affichée. Dans les cas classiques d'utilisation de menus ou de boutons pour l'affichage des listes de mots, l'utilisateur n'a pas de feedback sur les extrémités de la liste, il doit donc effectuer un geste ou un clic supplémentaire pour s'assurer de la fin de la liste (dans le cas par exemple où le mot désiré ne figure pas parmi ceux proposés par le système). Dans UniGlyph, la ZAS est manipulée à la façon d'une roulette avec un geste horizontal pour faire « tourner » son contenu.

4.1.1.2. Les modes du clavier

L'interface décrite à la Figure III-9 est l'interface principale du clavier UniGlyph. Elle correspond au mode de saisie de mots comme indiqué dans la zone de feedback à gauche de la touche de commande. Le clavier virtuel UniGlyph dispose de cinq modes différents : quatre modes de saisie respectivement des mots, lettres, chiffres et ponctuation et un mode de commande qui permet de naviguer entre ces modes. L'interface est assez stable car seule la petite zone de feedback et le contenu affiché par la ZAS changent d'un mode de saisie à l'autre. Dans le mode de commande, par contre, les touches de primitives sont remplacées par des touches de fonctions (suppression, espacement, retour à la ligne) et la ZAS affiche les abréviations des noms des quatre modes de saisie pour y permettre l'accès.

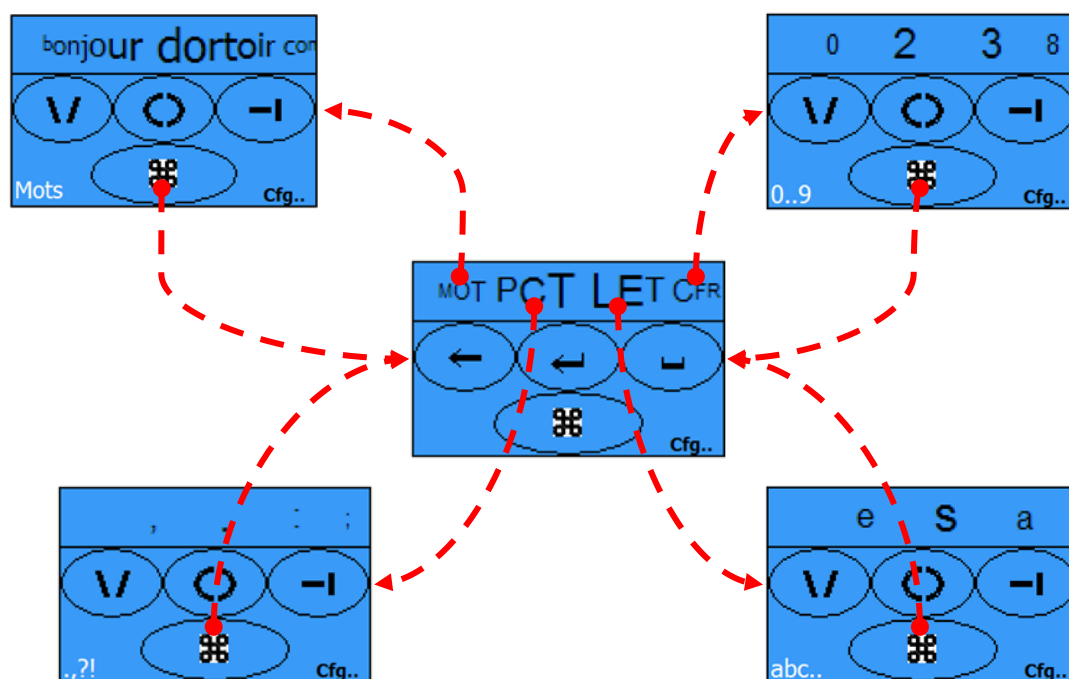


Figure III-11. Les différents modes du clavier UniGlyph.

La Figure III-11 montre les interfaces correspondantes aux différents modes de saisie et celle du mode de commande ainsi que l'architecture pour naviguer entre ces modes. Dans cette figure, chaque flèche montre qu'un appui sur l'élément au départ de la flèche permet d'accéder à l'interface désignée par cette dernière. Dans le mode de commande, l'appui sur la touche de commande permet de retourner au mode de saisie précédant. Les raccourcis affichés par la ZAS en mode de commande correspondent aux modes suivants :

a. Le mode de saisie de mots « MOT »

Permet à l'utilisateur de saisir un mot de longueur N (nombre de lettres) en effectuant une séquence de N appuis sur les touches de primitives. Par exemple, pour saisir le mot « bonjour » l'utilisateur doit appuyer sur la séquence de touches suivantes « 2-2-1-3-2-3-1 » (en considérant la numérotation des touches suivante : 1 pour la touche des traits inclinés, 2 pour la touche des traits arrondis et 3 pour la touche des traits droits). À chaque appui, la liste des mots correspondants à la séquence entrée apparaît sur la ZAS, triée selon les fréquences des mots dans la langue (cf. le système de désambiguïsation détaillé dans la section 4.1.2.2). Le premier mot de la liste proposée après chaque appui est inséré automatiquement comme feedback à la fin du texte déjà saisi. La liste de mots correspondants à la séquence d'appuis « 2-2-1-3-2-3-1 » est « bonjour, dortoir, contour, scalper ». L'utilisateur peut ensuite appuyer sur le mot désiré pour l'insérer. Si le mot désiré est le premier affiché sur la ZAS, il est inséré par défaut comme feedback grisé à la fin du texte saisi, l'utilisateur peut simplement insérer un espacement, un retour à la ligne ou une marque de ponctuation pour le valider. Cela n'est pas rentable dans le cas de l'insertion d'un espace car ce dernier correspond à deux appuis (un premier sur la touche de commande et un deuxième sur la touche d'espacement « _ ») alors qu'un seul appui pour choisir le mot dans la ZAS permet de le valider et d'insérer automatiquement un espacement après. En revanche, dans le cas où le mot est suivi d'un retour à la ligne ou d'une marque de ponctuation, cette fonctionnalité permet d'économiser un appui.

Un problème qui peut se poser est que, lors de la saisie d'un mot, l'utilisateur ne sait plus à quelle lettre il en est. Le feedback de la saisie (le mot grisé affiché par défaut) l'informe sur le nombre de lettres qu'il a tapées ce qui lui permet de savoir quelle lettre il doit entrer à la suite. Cette technique de feedback est utilisée par toutes les méthodes de saisie ambiguës. Une autre solution que nous avons étudiée consiste à afficher en feedback (au lieu du premier mot prédit) la séquence de primitives entrée. Cette option pourrait éviter les éventuels temps morts lors de la saisie pour se repositionner dans le mot à saisir mais son application n'est pas évidente : si on affiche une primitive par appui, la question qui se pose est laquelle choisir parmi les deux correspondantes à la touche appuyée ? Même en ajoutant un système de prédiction de primitive, la séquence affichée ne peut pas correspondre toujours exactement au mot que l'utilisateur désire saisir (voir Figure III-12-b). La deuxième solution peut consister à afficher les deux primitives de la touche appuyée à chaque appui, cela peut être perçu par l'utilisateur comme deux lettres successives et perturberait par conséquent l'utilisateur au lieu de lui faciliter la saisie. La troisième idée consiste à combiner les deux primitives de chaque appui en un seul symbole, ces nouveaux symboles peuvent être perçus comme de nouvelles primitives et ainsi augmenter la charge mentale de l'utilisateur pour les

déchiffrer (voir Figure III-12-c). Dans notre clavier actuel, nous avons opté pour la technique classique qui consiste à afficher le premier mot de la liste de désambiguïsation (comme avec T9).

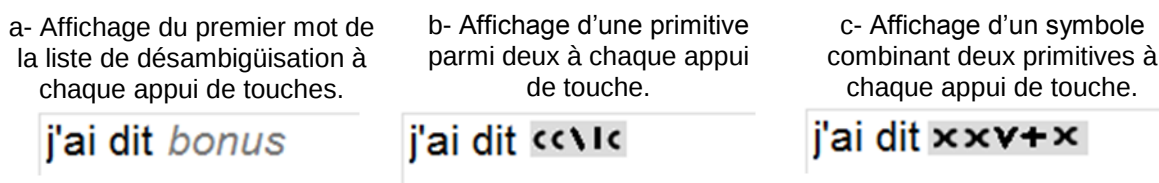


Figure III-12. Le feedback lors de la saisie. Exemple de la saisie de la phrase « j'ai dit bonjour », le mot en cours de saisie est « bonjour », l'utilisateur a déjà effectué 5 appuis sur les touches de primitives, il lui reste à effectuer les appuis correspondants aux lettres 'u' et 'r'.

Les exemples cités ci-dessus montrent uniquement des mots composés de lettres non accentuées mais UniGlyph permet également de saisir de la même façon les mots qui contiennent d'autres types de caractères.

Pour les ligatures (œ...) et les caractères diacritiques (À, Ê, Ç, ...), la saisie se fait comme pour les lettres sur lesquelles elles sont basées. Par exemple, le caractère 'Ê' correspond à la touche « —l » comme la lettre 'E'. Les ligatures sont donc traitées comme deux lettres séparées, et les caractères diacritiques sont saisis en un seul clic contrairement à la majorité des autres méthodes qui permettent de les saisir :

- en deux temps comme deux caractères successifs (Graffiti, Block Recognizer),
- en effectuant un ou plusieurs clics supplémentaires après la saisie du caractère de base (Glyph2, clavier téléphonique T12),
- en basculant en un mode dédié à ces caractères (clavier virtuel AZERTY).

Pour l'apostrophe, elle est attachée au mot qui la précède et correspond à un trait incliné. Par exemple, la conjonction « que » suivie d'une apostrophe (qu') est saisie en tapant la séquence de touches suivante : « C) » pour 'q', puis « —l » pour 'u' et « \ / » pour l'apostrophe.

b. Le mode de saisie des lettres « LET »

L'utilisateur peut avoir besoin de saisir des mots non communs, ou qui ne sont pas disponibles dans le dictionnaire principal d'UniGlyph. Dans ce cas, il peut basculer en mode de saisie des lettres pour entrer le mot désiré, caractère par caractère. Dans ce mode, la ZAS affiche les lettres triées par fréquences de bigrammes dans la langue. L'utilisateur peut appuyer sur l'une des touches de primitives pour afficher uniquement les lettres correspondantes à cette touche. Il peut ensuite faire défiler les lettres affichées par un geste horizontal sur la ZAS pour sélectionner une lettre comme dans le mode de saisie de mots.

Quand l'utilisateur entre un espacement ou tout autre caractère séparateur, le système vérifie si le mot saisi existe dans le dictionnaire. Si ce n'est pas le cas, une fenêtre s'affiche pour demander à l'utilisateur s'il veut ajouter ce mot au dictionnaire. À la validation, une deuxième fenêtre s'affiche pour lui permettre de situer ce mot par

rapport à d'autres lors de l'affichage de mots dans la ZAS (voir Figure III-13). Ces mots sont bien sûr ceux qui correspondent à la même séquence de touches que le mot ajouté. En fonction de la position choisie par l'utilisateur, une fréquence d'occurrence est déduite à partir des mots voisins et la nouvelle entrée est ajoutée dans un dictionnaire dédié (voir détails plus loin dans la section détaillant la prédiction).

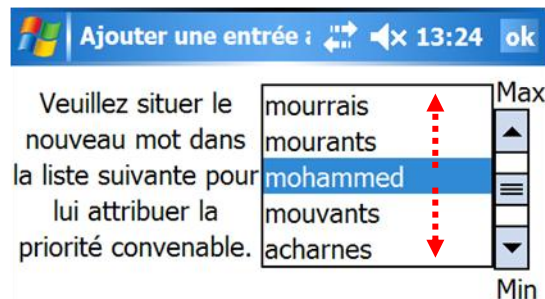


Figure III-13. Fenêtre d'ajout de nouveaux mots dans le dictionnaire. Exemple de l'entrée « mohammed ».

c. Le mode de saisie des chiffres « CFR »

Dans ce mode, les chiffres s'affichent sur la ZAS et peuvent être filtrés par des appuis sur les touches de primitives comme dans le mode de saisie des lettres. Dans une première version qui a été publiée en 2007 [Belatar et Poirier, 2007b], chaque chiffre correspond à une ou deux primitives pour reconstituer sa forme de la même façon que dans Glyph 2 (voir Chapitre II) mais en utilisant le jeu de primitives d'UniGlyph. La Figure III-14 montre cette décomposition des chiffres en primitives UniGlyph. Dans l'exemple de la Figure III-11, l'utilisateur a effectué un appui sur la touche des traits arrondis.



Figure III-14. Représentation des chiffres en une à deux primitives.

À la saisie, l'utilisateur peut effectuer un premier appui sur une touche de primitives pour que la ZAS affiche seulement les chiffres qui commencent par ces primitives. Par exemple, si la touche appuyée est celle des traits inclinés, les chiffres affichés sur la ZAS sont '4' et '6'. L'utilisateur peut ensuite en choisir un en le désignant sur la Zone d'Affichage et de Sélection ou appuyer sur la touche correspondante à la deuxième primitive. Dans l'exemple précédant, si l'utilisateur appuie sur la touche des traits droits (après avoir appuyé sur la touche des traits inclinés) le chiffre saisi est '4'. Dans cette configuration, certains caractères qui correspondent à la même séquence de touches ('3', '8' et éventuellement '0') ne peuvent pas être validés en entrant la deuxième primitive. L'utilisateur doit absolument choisir le chiffre désiré à partir de la ZAS soit après avoir entré la première primitive, soit après la deuxième. Cela donne donc des représentations différentes des chiffres (une ou deux primitives) et des façons différentes de les saisir : une sélection directe à partir de la ZAS, un appui de touche de primitives puis une sélection à partir de la ZAS, deux appuis de touches ou deux appuis de touches et une sélection. Cette inhomogénéité dans la saisie des chiffres peut constituer une difficulté pour l'utilisateur et ainsi rendre difficile la saisie des chiffres.

Une deuxième version a été développée ensuite. Elle consiste à utiliser le même principe de choix de primitives que pour les lettres. Chaque chiffre est représenté par une seule primitive (voir Figure III-15) selon la même règle de la saisie UniGlyph que nous reprenons ici pour les chiffres :

- Si le dessin du chiffre contient un trait incliné, alors appuyer sur la touche des « traits inclinés »,
- Sinon, s'il contient une boucle ou un arrondi, alors appuyer sur la touche des « traits arrondis »,
- Sinon appuyer sur la touche des « traits droits ».

Ce premier appui permet de filtrer seulement les chiffres correspondants à la touche appuyée sur la zone d'affichage ZAS. Un deuxième appui est nécessaire pour sélectionner le chiffre désiré à partir de cette zone.



Figure III-15. Représentation des chiffres en une seule primitive.

d. Le mode de saisie des marques de ponctuations « PCT »

Dans ce mode, toutes les marques de ponctuation sont affichées sur la ZAS et peuvent être défilées et sélectionnées comme les lettres, les chiffres et les mots (défilement par geste horizontal et sélection par simple appui). Le point et la virgule, qui sont les marques de ponctuation les plus utilisées lors de la saisie de texte, sont accessibles à partir de tous les autres modes en effectuant un appui long sur la touche de commande. Cela permet d'éviter les appuis supplémentaires qui permettent d'accéder au mode de ponctuation et de revenir au mode de saisie utilisé.

Dans sa version actuelle, le clavier UniGlyph offre donc quatre modes de saisie. D'autres modes peuvent être ajoutés pour étendre les fonctionnalités du clavier ou pour ajouter des cartes de caractères spéciaux par exemple. Pour chaque mode ajouté, doit correspondre un raccourci sur la ZAS en mode de commande.

4.1.1.3. La configuration du clavier

Le menu « Cfg.. » à la droite de la touche de commande donne accès à une fenêtre de configuration. Dans la suite, nous citons les principales options de configuration offertes par cette fenêtre.

a. Geste vertical pour les majuscules

Une fois cette option activée, la ZAS détecte, en plus du geste horizontal pour le défilement, le geste vertical (vers le haut) pour la mise en majuscule de la première lettre. L'utilisateur peut donc, au lieu de faire un simple clic pour valider un mot, pointer ce dernier avec le stylet et effectuer un geste vers le haut pour que le mot soit saisi débute par une lettre en majuscule.

b. Geste horizontal pour désigner une seule primitive

Cette option permet de réduire l'ambiguïté du clavier. En appuyant sur une touche de primitives, l'utilisateur peut désigner l'une des deux primitives de cette touche en

effectuant un geste vers la droite ou vers la gauche. Cela permet d'avoir une liste de mots candidats moins longue et éviter éventuellement de défiler la ZAS pour chercher le mot désiré. L'ambiguïté du clavier est réduite de moitié. Un geste sur une touche de primitives correspond à une moyenne de 4,33 lettres alors qu'un simple appui correspond à une moyenne de 8,66 lettres possibles. L'utilisation du geste réduit cependant la vitesse de la saisie. D'autres méthodes ont opté pour l'utilisation de la sélection gestuelle d'un élément parmi plusieurs sur une touche ambiguë. Par exemple, TenGo¹ utilise cette technique pour ajouter de nouveaux mots dans le dictionnaire, la version logicielle du clavier MessageEase² [Nesbat, 2003] utilise jusqu'à huit directions gestuelles sur la même touche pour entrer les lettres les moins fréquentes.

D'autres idées pour réduire l'ambiguïté du clavier ou pour faciliter l'apprentissage peuvent être étudiées. L'utilisation de pie-menus peut être l'une de ces idées. À l'appui sur une touche de primitives, les lettres correspondantes s'afficheront autour pour permettre d'en sélectionner une avec un simple geste. Plusieurs méthodes de saisie comme Claviature³ ou SpeedScript⁴ utilisent cette technique mais nous l'avons écartée pour le moment pour rester dans le cadre d'une méthode qui peut être implémentée sur différents types de dispositifs (versatility) et donc qui ne doit pas inclure dans ses principes de base des techniques contraintes par l'utilisation du tactile ou du gestuel.

c. Gestes sur la touche de commande

Toujours dans la même idée d'exploiter le geste de façon facultative dans cette implémentation d'UniGlyph, cette option permet d'effectuer les trois commandes de base dans l'édition de texte (à savoir, la suppression, le retour à la ligne et l'espacement) en un simple geste analogique et métaphorique par rapport à la présentation classique de ces caractères sur un clavier au lieu de deux appuis dans la configuration de base (en passant par le mode de commande). L'espacement correspond donc à un geste partant de la touche de commande vers la droite, le retour à la ligne correspond à un geste de la touche de commande vers le bas ou incliné vers le bas et la gauche et la suppression correspond à un geste partant de la touche de commande vers la gauche. Dans ce dernier cas nous avons intégré deux types de suppressions : la suppression d'un caractère ou l'annulation d'une primitive avec un geste court et la suppression d'un mot avec un geste long (voir Figure III-16). Puisque la touche de commande est assez large, sa bordure est utilisée comme limite entre le geste court et le geste long. En d'autres termes un geste complètement à l'intérieur de la touche de commande est défini comme un geste court et un geste qui dépasse le bord de la touche de commande est défini comme un geste long. Cette différenciation n'est pas prise en compte dans les cas de l'espacement et du retour à la ligne, seule la direction du geste est considérée.

¹ TenGo de Xrgonomix Inc., site web : www.tengo.net/, visité le 12/11/2008.

² MessageEase de EXideas, site web : www.exideas.com/ME/, visité le 12/11/2008.

³ Claviature de MICROTH Inc., site web : www.microth.com, visité le 15/10/2008.

⁴ SpeedScript, site web : www.speedscript.biz/, visité le 12/11/2008.

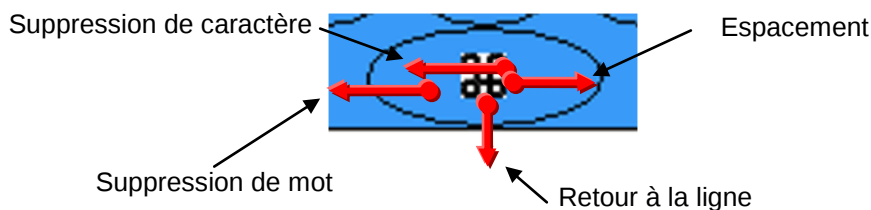


Figure III-16. L'option des commandes gestuelles.

Une autre idée que nous avons explorée consiste à reconnaître les gestes correspondants aux commandes sur toute la surface du clavier. Cette idée a été abandonnée car elle nécessite d'intégrer un module de reconnaissance des gestes ou d'en concevoir un qui soit adapté. Cela implique dans les deux cas une utilisation supplémentaire des ressources du dispositif mobile, chose que nous voulons éviter pour pouvoir exploiter ces ressources dans un système de prédiction assez « performant ».

d. Appuis longs sur les touches

L'activation de cette option permet de reproduire la fonction d'une touche plusieurs fois d'une façon rapide en appuyant en continu sur la touche concernée. Elle est utile surtout en mode de commande pour effectuer une succession de suppressions en continu par exemple.

Sur la touche de commande, l'appui long permet d'accéder à d'autres fonctions que la suppression, le retour à la ligne et l'espacement. Les trois touches correspondantes à ces fonctions en mode de commande sont : la suppression d'un mot entier, la saisie du point et de la virgule comme le montre la Figure III-17. Cela évite de passer par le mode de ponctuation pour entrer les marques de ponctuation les plus fréquemment utilisées.

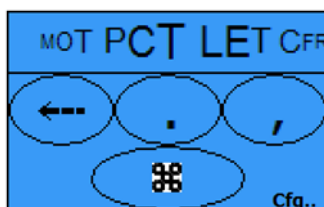


Figure III-17. Le mode de commande accédé par un appui long sur la touche de commande.

e. Exploitation du pavé directionnel

La grande majorité des assistants numériques disposent d'un pavé de cinq boutons appelé « le pavé directionnel ». Il est composé de quatre boutons directionnels (droite gauche, haut et bas) et d'un bouton dit de validation (Ok). Le clavier UniGlyph pour PDA peut exploiter ce pavé directionnel. Les boutons directionnels remplacent les trois touches de primitives et le bouton de validation remplace la touche de commande dans UniGlyph. Seule la ZAS reste affichée sur l'écran (voir Figure III-18). Avec cette configuration, le clavier UniGlyph occupe encore moins d'espace sur l'écran et laisse une plus grande visibilité sur l'application qui est derrière et sur le texte saisi.

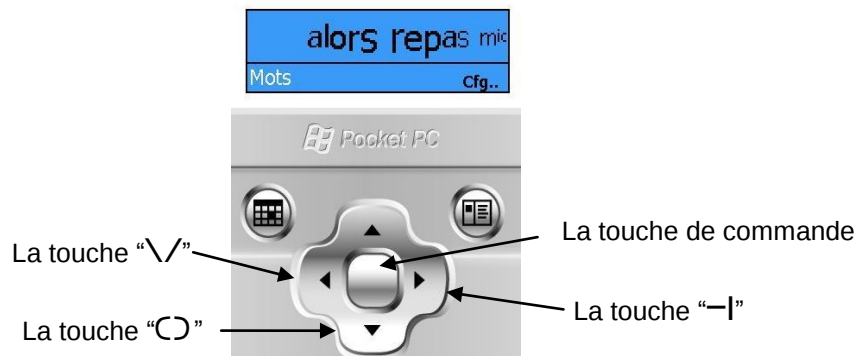


Figure III-18. Utilisation des touches physiques du pavé directionnel.

La figure ci-dessus montre le clavier UniGlyph en mode de saisie de mots lorsque l'option d'utilisation du pavé directionnel est activée. Cette configuration est destinée aux utilisateurs bien familiarisés avec le clavier UniGlyph puisque les boutons ne sont pas étiquetés par les primitives. En mode de commande le bouton « gauche » correspond à la suppression, le bouton « bas » correspond au retour à la ligne et le bouton « droite » correspond à l'espace.

f. Affichage des lettres sur les touches

L'avantage principal de l'utilisation du principe de la saisie ambiguë à primitives est d'avoir un nombre minimal de touches avec un nombre minimal de libellés pour éviter l'encombrement de l'interface et réduire le temps de la recherche visuelle. Les méthodes de saisie ambiguë à très peu de touches étiquetées par les lettres, comme celle décrite par Dunlop [2004] (méthode GORSUV pour la saisie de texte sur montre), sont souvent difficiles à utiliser à cause de la distribution purement statistique des lettres sur les touches et à cause de l'encombrement de l'interface. Cependant, une méthode de saisie à touches sans labellisation en lettres peut aussi être difficile à utiliser dans une première phase d'apprentissage quand le principe de l'analogie n'est pas encore maîtrisé. Pour cela, UniGlyph donne à l'utilisateur la possibilité d'afficher les lettres correspondantes à chaque touche autour des primitives comme le montre la Figure III-19.



Figure III-19. Affichage des lettres sur les touches de primitives.

Cette option permet de solliciter l'utilisateur de deux façons différentes. Avec des touches non étiquetées des lettres, il s'appuie sur la connaissance de la méthode et puise ses « connaissances dans la tête », avec des touches étiquetées, il scrute le clavier et puise ses « connaissances dans le monde » [Norman, 1980].

g. Proposer une liste de complétion

La liste de mots affichés sur la ZAS qu'on a décrit jusqu'ici correspond à une liste de « désambiguïsation ». En d'autres termes, elle contient seulement les mots qui correspondent exactement au nombre d'appuis de touches effectués par l'utilisateur. UniGlyph propose une autre option qui permet d'afficher une liste de mots plus longs qui

commencent par la séquence d'appuis effectués. L'utilisateur peut activer cette option et choisir le nombre de mots à proposer dans cette liste que nous appelons « liste de complétion ». Nous avons choisi de limiter ce choix dix mots au maximum (l'utilisateur peut choisir que cette liste contienne 1, 2, 3...ou 10 mots) car cette liste est affichée à la fin de la liste de désambiguïsation sur la ZAS et n'est accessible donc, qu'après le défilement de tout le contenu de cette dernière. La liste de complétion est utile surtout lors de la saisie d'un mot long où la liste de désambiguïsation devient très réduite, voire vide. Les mots de la deuxième liste sont donc plus accessibles dans ce cas et permettent de choisir le mot désiré sans effectuer la suite d'appuis nécessaires. Par exemple, pour saisir le mot « mayonnaise », après la saisie des six premières primitives, le mot « mayonnaise » s'affiche en deuxième position sur la ZAS car la liste de désambiguïsation ne contient qu'un seul mot (voir les détails du système de désambiguïsation et de prédiction plus loin).

4.1.2. L'architecture logicielle du clavier

UniGlyph est développé sous forme de composant .NET et donc distribué sous forme de DLL. Nous avons choisi d'utiliser le langage de programmation C# pour plusieurs raisons dont l'aspect objet, la facilité et la rapidité du codage et surtout pour la grande compatibilité et robustesse du .NET Framework avec Windows Mobile, le système d'exploitation le plus utilisé sur les assistants numériques (PDA) au début de cette thèse avec 52,6% de part de marché contre 25,5% pour RIM OS, 13,4% pour Palm OS, 3,6% pour Symbian et 1,2% pour Linux selon les statistiques de l'agence de recherche et de consultation en technologies d'information Gartner¹ en mai 2006.

Pour le codage en C# nous avons utilisé l'environnement de développement Visual Studio .NET 2005 de Microsoft. Les déploiements et les tests ont été réalisés sur un Pocket PC DELL Axim 51x avec 624 MHz d'horloge processeur, 64 Mo de mémoire d'applications et 256 Mo de mémoire de stockage. L'écran est de type VGA de 3,7 pouces.

La DLL UniGlyph s'appelle UniGlyphLib. Elle propose des interfaces simples et claires pour être exploitée à partir de toute application Windows Mobile. Dans la suite nous présentons l'architecture générale de cette DLL en mettant l'accent sur ces interfaces et en détaillant l'un des éléments les plus importants dans cette architecture, à savoir, le système de désambiguïsation et de prédiction.

4.1.2.1. L'architecture générale

Le schéma de la Figure III-20 montre les grands traits de l'architecture de la DLL UniGlyphLib.

¹ <http://www.gartner.com/it/page.jsp?id=492420>, page web visitée le 11/10/2008.

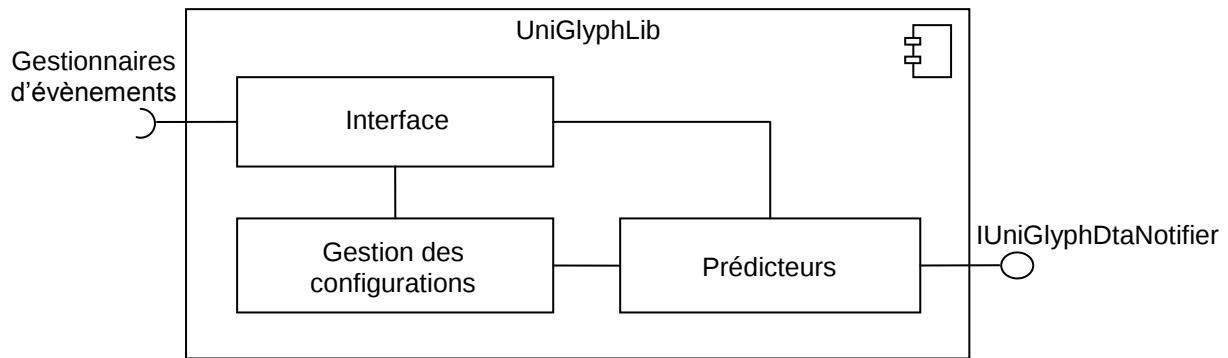


Figure III-20. Architecture globale de la DLL UniGlyphLib.

La partie « interface » est constituée essentiellement des contrôles correspondants aux touches (ImageButton) et à la ZAS (ZoomList) et de la classe principale de l'interface du clavier (Keyboard). La partie « prédicteurs » est constituée d'un prédicteur de mots (WordPredictor), d'un prédicteur de lettres (CharPredictor) et d'un gestionnaire d'accès et de modification des données linguistiques sur le système de fichiers (RandomAccessDatabase). Le diagramme de classes complet est détaillé dans l'annexe 4.

Dans cette section nous mettons l'accent sur l'interface IUniGlyphDtaNotifier qui permet à l'application d'édition de texte de communiquer avec le composant UniGlyph.

Code C#
<pre> public interface IUniGlyphDtaNotifier { string LastWord {get;} void ResolvedWord(string word); void SuggestedWord(string word); void PrimEntered(string primitive); void CharEntered(string character); void NumEntred(string no); void PunctMarkEntered(string pm); void CommandExe(Commands cmd); void RefreshPred(); } </pre>

Tableau III-6. Structure de l'interface de communication de l'application avec le clavier.

Le développeur de l'application mère qui intègre UniGlyph doit implémenter cette interface et définir l'ensemble de ses membres qui ont les fonctions suivantes :

- **LastWord** : retourne une chaîne de caractères qui correspond au mot qui précède le curseur de la saisie dans la zone d'édition en cours (e.g. TextBox). Cela permet au clavier de récupérer le dernier mot après la saisie d'un espace en mode de saisie des lettres pour vérifier par exemple s'il existe dans le dictionnaire et proposer de l'ajouter au dictionnaire le cas échéant.
- **ResolvedWord(string)** : notifie à l'application que l'utilisateur a validé un mot à partir de la zone d'affichage et de sélection du clavier UniGlyph. La définition de

cette méthode doit généralement permettre d'insérer ce mot à l'emplacement du curseur dans la zone d'édition et d'insérer un espace à la suite.

- `SuggestedWord(string)` : permet de passer à l'application le premier mot affiché sur la ZAS après chaque action de l'utilisateur sur le clavier UniGlyph. Le développeur de l'application qui utilise UniGlyph peut définir cette méthode de façon à insérer ce mot, grisé par exemple, à l'emplacement du curseur de la saisie (comme dans le cas de la Figure III-12-a).
- `PrimEntered(string)` : notifie à l'application que l'utilisateur a entré une primitive en la passant en argument. Cela permet d'afficher directement les primitives dans la zone d'édition si le développeur de l'application ne souhaite pas afficher le premier mot de la liste de prédiction.
- `CharEntered(string)` : notifie à l'application que l'utilisateur a validé une lettre à partir de la ZAS.
- `NumEntered(string)` : notifie à l'application que l'utilisateur a entré un chiffre.
- `PunctMarkEntered(string)` : notifie à l'application que l'utilisateur a entré une marque de ponctuation. La définition de cette méthode doit gérer aussi l'insertion des espacements selon la marque de ponctuation insérée (un espace avant les deux points ':', un espace après la virgule ',' et des espaces avant et après le point-virgule ';' par exemple).
- `CommandExe(UniGlyphLib.Commands)` : permet au clavier de transmettre les quelques commandes d'édition de texte à l'application qui l'implémente. La définition de cette méthode doit gérer le traitement de ces commandes. Ces dernières sont transmises en argument de type « `UniGlyphLib.Commands` » qui est une énumération des valeurs suivantes :
 - `UniGlyphLib.Commands.DEL` : pour l'annulation de la dernière primitive entrée ou, si aucune primitive n'a été entrée, la suppression du dernier caractère du texte saisi.
 - `UniGlyphLib.Commands.LNGDEL` : pour l'annulation de la séquence de primitives entrées ou, dans le cas échéant, la suppression du dernier mot saisi.
 - `UniGlyphLib.Commands.ENTER` : pour insérer un retour à la ligne et éventuellement valider le dernier mot suggéré à l'utilisateur (le premier mot de la liste de prédiction proposé en grisé à la fin du texte saisi dans les exemples précédents).
 - `UniGlyphLib.Commands.SPACE` : pour insérer un espacement et éventuellement valider le dernier mot suggéré à l'utilisateur.

À part la méthode « `CommandExe(Commands)` », les valeurs transmises à l'application par les autres méthodes sont de type chaîne de caractères pour simplifier leur insertion dans le texte édité.

4.1.2.2. La désambiguïsation et la prédiction

Le module de désambiguïsation et de prédiction de mots est chargé de proposer à l'utilisateur une liste de candidats qui correspondent exactement, ou dont le début correspond à la séquence d'appuis effectuée à chaque instant. La désambiguïsation dont

on parle ici est une désambiguïsation lexicale, autrement connue sous le nom de « Dictionary-Based Disambiguation ». Elle est combinée à une prédiction lexicale à base d'un modèle de langage uni-gramme (dictionnaire de fréquences des mots) pour fournir une liste de candidats où le mot que l'utilisateur désire saisir, figure parmi les premiers.

L'utilisation d'un simple dictionnaire de fréquences (modèle uni-gramme) est la technique la plus adoptée dans les entrées de texte prédictives pour dispositifs mobiles [Dunlop et Crossan 2000]. Cela est dû certainement au fait que c'est le modèle qui utilise le moins de ressources. Cependant, un système de prédiction plus performant pourrait nous assurer de bons résultats en réduisant le besoin de défiler la ZAS pour atteindre les mots à saisir. D'autres travaux de recherche ont essayé d'implémenter de tels prédicteurs en négligeant la contrainte matérielle et technique soit en mettant l'hypothèse que les dispositifs mobiles sont en pleine croissance et que le problème matériel ne doit pas être un frein, soit en mettant des hypothèses de solutions « potentielles » comme la centralisation et l'accès à distance aux données linguistiques grâce aux technologies de transfert de données comme le GPRS et la 3G [Hasselgren et al., 2003].

De notre part, nous avons choisi d'implémenter une solution fonctionnelle et performante tout en restant dans les limites des contraintes matérielles actuelles sachant que notre appareil de test et de simulation est l'un des plus performants sur le marché à l'heure actuelle (un PDA avec 624 MHz de processeur, 64Mo de mémoire d'application et 256Mo de mémoire de données).

Le dictionnaire actuel d'UniGlyph contient les 41 000 mots les plus fréquents extraits de la base LEXIQUE¹ selon les fréquences des mots dans un corpus de sous-titres de films [New et al., 2007]. Nous avons appliqué les traitements suivants sur le dictionnaire de LEXIQUE3 pour l'adapter à notre cas et pour réduire sa taille :

- Suppression de toutes les entrées non lexicales (comme « zzz ») par intersection avec le dictionnaire de mots communs de l'ABU².
- Suppression de tous les mots composés.
- Conversion des fréquences de mots en entiers.

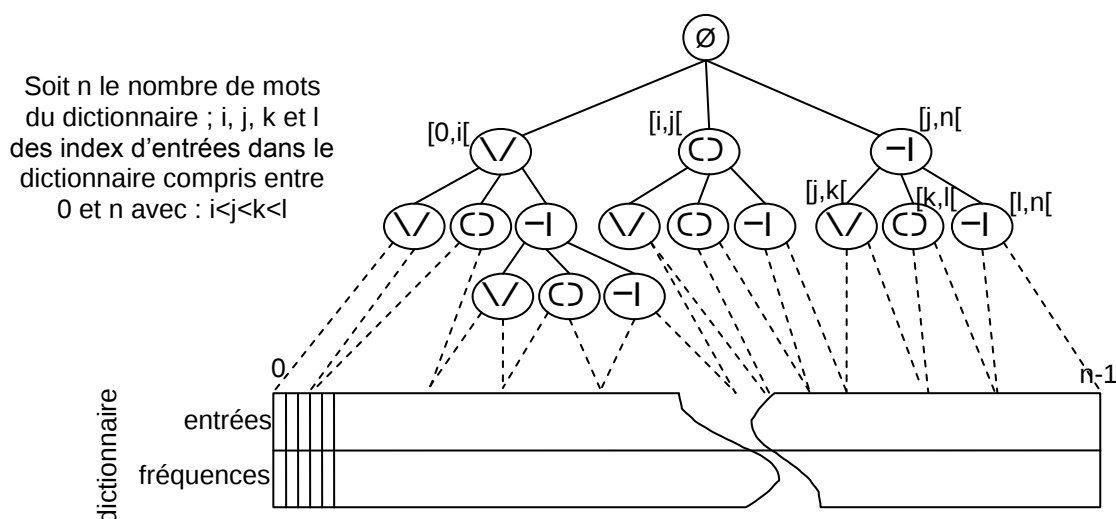
Nous avons choisi les fréquences des mots obtenues à partir d'un corpus de films plutôt que les fréquences des livres car les premières sont les plus proches des conditions d'utilisation d'UniGlyph, à savoir, la saisie de texte sur dispositifs d'infocommunication mobiles.

Le module de prédiction dans UniGlyphLib est composé essentiellement de la classe « WordPredictor » qui assure la génération des candidats correspondant à une séquence de primitives donnée et « RandomAccessDatabase » qui lui, permet l'accès aux fichiers de données (voir annexe 4). Au lancement du clavier, le prédicteur charge le dictionnaire de

¹ <http://www.lexique.org>

² <http://abu.cnam.fr/DICO/>

mots en mémoire sous forme d'arbre de segments comme l'illustre la Figure III-21. Cette solution nous permet de charger un large dictionnaire en utilisant un minimum de mémoire. Elle nous permet aussi un stockage binaire des fichiers sans données supplémentaires et un accès directe et rapide en lecture et en écriture pour la modification (update). L'inconvénient de cette solution est que le temps de chargement des données en mémoire est très important (de l'ordre d'une minute).



Une autre solution que nous avons étudiée consiste à stocker l'arbre sous format XML et d'y accéder par un parseur de la famille SAX (Simple API for XML). Cette solution résout le problème du délai de chargement des données mais augmente considérablement la taille de données stockées à cause de l'ajout des balises et du stockage des fréquences des mots sous format textuel. Un autre inconvénient de cette solution réside dans la lenteur du parcours de l'arbre avec le parseur « XMLReader » (équivalent .NET de SAX) surtout en Compact Framework et sur un dispositif à faibles ressources. De plus, cette technique ne facilite pas l'accès en écriture.

Avec la solution du stockage binaire du dictionnaire et la création dynamique de l'arbre, seul un index est rajouté au dictionnaire de mots-fréquences. Le dictionnaire est divisé donc en trois colonnes : index, mots et fréquences. Il est trié par ordre de correspondance aux séquences d'appuis puis par ordre décroissant des fréquences pour les mots qui correspondent à une même séquence. Cela facilite la construction de l'arbre de segments et permet de fusionner la prédiction avec la désambiguïsation en évitant de trier la liste de mots résultante après la désambiguïsation. Chaque colonne est divisée en trois parties (sur trois fichiers) pour faciliter la gestion et l'accès aux données. Cela donne neuf fichiers avec une taille totale de 624 Ko répartie de la façon suivante :

- Les fichiers d'index contiennent les index des 41427 mots dans les fichiers binaires de mots. Ces index sont codés en « UInt16 » (entier non signé sur 2 octets). Cela donne une taille totale de 81 Ko.
- Les fichiers de fréquences contiennent la fréquence correspondante à chaque mot codée en « UInt32 » avec une taille totale de 162 Ko.

- Les fichiers de mots contiennent les mots sous forme de séquences d'octets (Byte) délimitées par un entier qui contient la longueur du mot. La taille totale de ces fichiers est de 381 Ko.

Lors de la saisie du texte, chaque appui de touche, permet d'accéder à un nœud inférieur et obtenir une plage de mots dans le dictionnaire qui correspondent à la séquence d'appuis effectués. Cette plage de mots est composée dans l'ordre d'une liste de mots correspondant exactement à la séquence d'appuis effectués et d'une liste de mots plus longs qui commencent par cette séquence. La première liste est préalablement triée par ordre décroissant des fréquences comme nous l'avons précisé précédemment. La deuxième ne l'est pas puisqu'elle correspond à des séquences d'appuis différentes et de longueurs différentes (même si elles ont un préfixe commun qui est la séquence d'appuis effectuée par l'utilisateur). Si l'utilisateur a configuré son clavier UniGlyph de façon à afficher un nombre N de mots plus long à la fin de la ZAS (voir section 4.1.2.2), les N mots les plus fréquents de cette deuxième liste sont rajoutés à la fin de la première pour être proposés à l'utilisateur.

À part la prédiction basée sur un dictionnaire de fréquences, nous avons étudié plusieurs autres possibilités en commençant par le système de prédiction de la première version de Sibylle appelé SibyMot. Ce prédicteur intègre la notion de Chunk pour la prédiction des mots et permet ainsi un taux d'économie de saisie KSR (Keystroke Saving Rate) d'environ 57,1% [Schadle et al., 2004]¹. Après l'étude de SibyMot et de son implémentation, nous avons écarté cette possibilité à cause de sa complexité. Cette complexité se traduit par un prédicteur qui nécessite une grande quantité de ressources tant en traitement qu'en stockage de données. Sans prendre en compte la taille du programme lui-même et la quantité de mémoire nécessaire pour son exécution, SibyMot utilise 31,3 Mo de données (modèle de langage et données diverses utilisées par le prédicteur) qui doivent être déployées sur le terminal mobile. Cette solution est inexploitable sur un dispositif mobile à faibles ressources. Il ne serait pas acceptable que l'outil de saisie de texte occupe la totalité de la mémoire de l'appareil. Nous l'avons donc aussitôt écartée.

Nous avons ensuite implémenté et étudié des modules de désambiguïsation et de prédiction à base de modèles trigramme et bi-classe simplifiés (de façon à optimiser la taille des données stockées en mémoire). On rappelle ici que le modèle trigramme repose sur l'hypothèse simplificatrice que, étant donnée une séquence de 3 éléments, la probabilité de l'apparition d'un élément en troisième position ne dépend que des deux éléments précédents. Un modèle n -gramme correspond en fait à un modèle de Markov d'ordre n où seules les n dernières observations sont utilisées pour la prédiction de l'élément (dans notre cas, le mot) suivant. Concernant le modèle bi-classe, ou n -classe en général, le principe est de grouper les mots en classes selon un critère bien défini et de prédire, selon le même principe du modèle n -gramme, la classe du mot de rang n à partir des classes des $n-1$ mots précédents.

¹ Dans un contexte de saisie non ambiguë.

Les deux modèles ont été générés à l'aide de l'outil SRILM (the SRI Language Modeling Toolkit) [Stolcke, 2002] sur un corpus d'articles du journal Le Monde des années 1998-1999 (environ 244 000 000 mots) avec un filtrage sur le vocabulaire des quelques 41000 mots utilisés dans notre prédicteur uni-gramme décrit précédemment.

Dans le modèle bi-classes généré avec l'outil SRILM, les classes ne sont pas syntaxiques mais des classes statistiques (aussi appelées classes lexicales) générées automatiquement à partir du contexte dans lequel apparaît chaque mot dans le corpus d'apprentissage. Les mots qui partagent le même contexte lexical sont regroupés dans la même classe et c'est le développeur qui choisit le nombre de classes dans lesquelles regrouper le vocabulaire. Plus le nombre de classes est important et plus la prédiction est efficace mais la taille des données augmente. Dans notre cas, nous avons fixé le nombre de classes à 500 après avoir testé avec des modèles de 100 et de 1000 classes. Le nombre de 500 classes statistiques est également retenu dans d'autres travaux comme [Perraud et al., 2003] qui préfèrent déjà les modèles n-classe aux modèles n-gramme à cause de la taille des données stockées sur l'appareil mobile.

Nous avons traduit les deux modèles (trigramme et bi-classes) en arbres XML et nous avons implémenté des prédicteurs utilisant le parseur « XMLReader » (principes de la famille SAX). L'ensemble des fichiers du modèle trigramme pèse 72,3 Mo sur le système de fichiers et ceux correspondants au modèle bi-classes occupent quant à eux 3,84 Mo en tenant compte des nombreuses optimisations qu'on a appliquées et qui réduiront certainement les performances de ces modèles par rapport à une implémentation normale non contrainte par la quantité de données stockées.

Malgré la grande quantité de données utilisées par chacun des deux systèmes de désambiguïsation et de prédiction, nous avons décidé de les tester aux côtés de notre prédicteur uni-gramme sur PDA en stockant les données respectives sur une carte mémoire flash. Nous avons développé donc une petite application sur PDA pour la simulation des trois systèmes. Pour cette simulation nous avons fixé les paramètres (hypotheses) suivants :

- À chaque appui de touche, le système génère une liste de désambiguïsation complète en plus d'une liste de 5 mots de complétion (des mots plus longs dont le début correspond à la séquence d'appuis effectués) pour calculer le temps de réponse du système.
- On suppose que la liste de complétion est affichée à l'utilisateur séparément de la liste de désambiguïsation et qu'il valide le mot désiré dès qu'il apparaît dans la liste. Cela nous permet d'évaluer le taux d'économie de saisie (KSR pour Keystroke Saving Rate).
- Si pour un mot donné, toute la séquence d'appuis correspondante est effectuée, il est sélectionné à partir de la liste de désambiguïsation et correspond donc à un KSR de 0%.

Le corpus de test est composé d'une liste de 105 phrases simples de mots qui existent tous dans le vocabulaire utilisé par les prédicteurs (cf. le corpus utilisé plus tard pour

l'évaluation du clavier UniGlyph avec des utilisateurs, en 4.2.2). Les résultats de la simulation sont transcrits sur le tableau ci-dessous.

Modèle	Taille des données	Position des mots dans la liste	KSR	Temps moyen de réponse
Uni-gramme	624 Ko	2,2	26,25 %	27 ms
Bi-classes	3,84 Mo	1,47	13,80 %	86 ms
Trigramme	72,3 Mo	1,55	30,25 %	556 ms

Tableau III-7. Résultats de la simulation des trois systèmes de prédiction.

Cette comparaison montre la grande différence en utilisation de ressources entre le modèle uni-gramme et les deux autres. La prédiction reste raisonnable avec un KSR de 26% même si la position du mot dans la liste est plus éloignée qu'avec les deux autres prédicteurs.

4.2. Évaluations

4.2.1. Prédiction théorique des performances

Théoriquement, UniGlyph permet de saisir toutes les lettres (accentuées ou non) en un seul clic par caractère (1 KSPC). L'espace est entré en deux clics comme nous l'avons expliqué précédemment, mais il est très rarement saisi car un espace est automatiquement inséré quand l'utilisateur choisit un mot dans la liste de prédiction. On peut donc considérer que l'espacement correspond à un simple clic sur la ZAS. On obtient donc un KSPC théorique de 1.

La vitesse de saisie peut être calculée en se basant sur les lois de Fitts et de Hick-Hyman [Soukoreff et MacKenzie, 1995]. Cependant la prédiction du temps de la sélection des mots à partir de la liste d'affichage ne peut pas être significative car elle dépend de la nature du texte saisi, et donc du rang du mot à saisir dans la liste de désambiguïsation (voir [Sad et Poirier, 2008] pour les performances des listes de désambiguïsation). Dans les calculs qui suivent, nous estimons les performances maximales et donc nous considérons que le mot à saisir apparaît toujours parmi les premiers sur la ZAS et peut être validé, suivi d'un espace, en un simple clic. En d'autres termes, un clic sur une touche de primitive correspond à la saisie d'une lettre et un clic sur la ZAS correspond à la saisie d'un espace.

En se basant sur les études faites par MacKenzie et Soukoreff [Soukoreff et MacKenzie, 1995] [MacKenzie, 1995], on peut modéliser le temps de mouvement (TM_{ij}) entre une touche i et une touche j de la manière suivante :

$$TM_{ij} = \begin{cases} TM_{rep} , & i = j \\ a + b \times \log_2 \left(\frac{A_{ij}}{L_j} + 1 \right), & i \neq j \end{cases} \quad (1)$$

Avec :

- TM_{rep} : temps moyen pour recliquer sur la même touche. Ce temps est estimé à 153 ms pour une saisie à stylet sur écran tactile [Soukoreff et MacKenzie, 1995],

- a et b : constantes (intercept et pente) dont les valeurs sont respectivement '0' et $\frac{1}{4,9}$,
- A_{ij} : distance (amplitude) entre le centre de la touche i et celui de la touche j ,
- L_j : largeur de la cible dans le sens du mouvement.

Les différentes amplitudes A_{ij} et les largeurs L_j peuvent être calculées à partir de la Figure III-22 qui montre les dimensions du clavier UniGlyph (exprimées en pixel sur une résolution d'écran de 140 x 320).

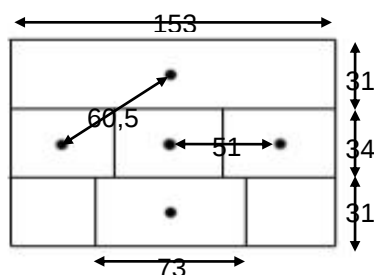


Figure III-22. Les dimensions du clavier UniGlyph.

Pour pouvoir calculer la vitesse de la saisie, il faut déduire le temps moyen du mouvement $\overline{TM}$ entre deux touches. Ce temps correspond à la saisie d'un caractère. Pour ce faire, il faut multiplier chaque TM_{ij} par la probabilité de la succession de la touche i par la touche j selon l'équation suivante :

$$\overline{TM} = \sum_{i,j \in \text{touches}} (P_{ij} \times TM_{ij}) \quad (2)$$

Succession de touches (un clic sur la ZAS correspond à l'espace)	Probabilité
même touche	0,20407595
« C) » puis « - »	0,035571293
« - » puis « ZAS »	0,077575405
« - » puis « \ / »	0,056444788
« \ / » puis « - »	0,038912054
« ZAS » puis « C) »	0,050926597
« - » puis « C) »	0,091056192
« ZAS » puis « - »	0,040533107
« \ / » puis « C) »	0,049756204
« C) » puis « \ / »	0,103212385
« C) » puis « ZAS »	0,096125465
« \ / » puis « ZAS »	0,094218933
« ZAS » puis « \ / »	0,061591627
Somme des probabilités :	1

Tableau III-8. Probabilités des successions de touches.

Le Tableau III-8 détaille les probabilités P_{ij} dans le cas du clavier UniGlyph. Ces valeurs ont été calculées à partir des fréquences des bigrammes du français, obtenues sur le corpus Corpatext1.02¹ qui contient environs 37 millions de mots.

À partir des équations (1) et (2), et en utilisant les données de la Figure III-22 et du Tableau III-8, on obtient : $\overline{TM} = 215,63$ ms. On peut donc déduire le nombre de caractères par seconde (CPS) et le nombre de mots par minute (WPM) de la manière suivante :

$$CPS = \frac{1}{(\overline{TM})} = 4,63 \quad (3)$$

$$WPM = \frac{CPS \times 60}{5} = 55,64 \quad (4)$$

Ces calculs ne tiennent compte que du temps de mouvement d'une touche à l'autre. En ajoutant une estimation du temps de la recherche visuelle ou « temps de réaction » selon la loi de Hick-Hyman [Soukoreff et MacKenzie, 1995], on obtient :

$$CPS = \frac{1}{(\overline{TM} + 0,2 \times \log_2(4))} = 1,63 \quad (5)$$

$$WPM = \frac{CPS \times 60}{5} = 19,56 \quad (6)$$

4.2.2. Évaluation utilisateur

4.2.2.1. Protocole de l'évaluation

L'évaluation des performances des méthodes d'entrée de texte a toujours mis en évidence de nombreuses questions sur la comparabilité des résultats obtenus pour chaque méthode. En effet, les chiffres publiés pour chaque méthode sont relatifs à un ensemble de paramètres qui ne sont malheureusement pas normalisés. Parmi ces paramètres :

- évaluation d'utilisateurs experts ou évaluation de la progression d'utilisateurs novices sur plusieurs sessions,
- utilisation d'un corpus de mots ou de phrases,
- utilisation de mots accentués ou non,
- utilisation d'un grand corpus avec sélection aléatoire ou utilisation d'un petit corpus (voire une seule phrase) avec la ressaisie des mêmes mots/phrases à chaque session,
- tâche de production (text creation task) ou tâche de recopie (text copy task),
- nombre de sessions, durée de chaque session, intervalle entre les sessions.

D'autres facteurs comme la langue, le lieu de l'évaluation, la position lors de la saisie et la manière de présenter l'évaluation et d'expliquer la méthode aux utilisateurs peuvent avoir une grande influence sur les résultats. Pour ces raisons, nous avons évité

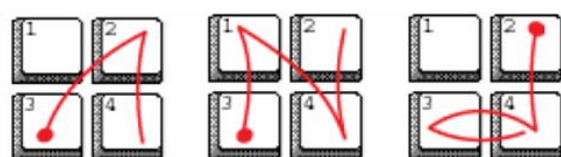
¹ Lexique 3, base de données linguistiques. Disponible à l'adresse <http://www.lexique.org/>, visitée le 21/09/2008.

dans le premier chapitre de faire un tableau comparatif des performances des méthodes citées.

Pour l'évaluation d'UniGlyph, nous avons choisi de faire une évaluation comparative avec deux autres méthodes dans le même environnement et avec le même protocole. En effet, Wobbrock et Myers [2006a] notent que la pratique actuelle est d'évaluer les nouvelles méthodes de saisie en comparaison à des méthodes existantes dans une expérience commune et contrôlée qui consiste à recopier des phrases le plus rapidement et le plus correctement possible (as quickly and accurately as possible). Cette pratique permet d'avoir une meilleure comparabilité des résultats.

Les deux méthodes retenues ont été sélectionnées de façon à être les plus proches possibles d'UniGlyph. Elles offrent des touches de grandes dimensions, permettant une éventuelle saisie au doigt.

La première est une variante à quatre touches de la méthode EdgeWrite [Wobbrock et al., 2006] que nous avons déjà présentée dans le premier chapitre. Elle repose, sur un principe de saisie par analogie aux caractères latins. Pour saisir un caractère, l'utilisateur doit reproduire sa forme sur les quatre touches qui sont disposées en carré comme le montre la Figure III-23-a.



a. Le clavier EdgeWrite à quatre touches



b. Le clavier Phrase-It®

Figure III-23. Les deux claviers évalués avec UniGlyph.

La deuxième, Phrase-It¹ (Figure III-23-b) est une méthode non académique qui repose sur une organisation alphabétique et permet une saisie en 2 clics par caractère. La configuration basique de Phrase-It donne un clavier de deux rangées de cinq touches chacune. La première ligne (de haut en bas) est composée de touches de commandes (Shift, Ponctuation, espace, suppression et retour à la ligne) et la deuxième contient les cinq voyelles (a, e, i, o et u). Quand une touche de voyelle est appuyée, les touches de la première ligne affichent les consonnes qui suivent cette voyelle dans l'ordre alphabétique. Par exemple, en appuyant sur la touche 'a', les touches de la première ligne affichent les lettres 'b', 'c', 'd'. Un deuxième appui est nécessaire pour sélectionner la lettre désirée. Si la lettre à saisir est une voyelle ('a', 'e', 'i', 'o' ou 'u'), elle est saisie donc en un double clic sur la même touche, sinon elle correspond à deux appuis sur deux touches différentes.

La première méthode (4-Key EdgeWrite) est comparable à UniGlyph du point de vue où elle utilise le même nombre de touches, qu'elle est basée sur un principe d'analogie aux caractères latins et que c'est une méthode conçue pour être indépendante du

¹ Prevalent Devices Inc., site web : www.prevalentdevices.com, visité le 21/09/2008.

dispositif. Elle présente cependant une grande différence au niveau du nombre d'appuis par caractère.

La deuxième méthode (Phrase-It) est comparable à UniGlyph dans le sens où elle constitue une alternative au clavier logiciel traditionnel qui vise à augmenter la taille des touches sans augmenter la taille du clavier afin de permettre une saisie plus rapide et une possibilité d'interaction au doigt. La différence principale avec UniGlyph réside dans le nombre de touches et dans le nombre d'appuis nécessaires pour entrer chaque type de caractères : alors qu'UniGlyph permet de saisir les lettres en un appui et les caractères de commandes (suppression, espacement, retour à la ligne) en deux appuis, Phrase-It fait le contraire en attribuant deux appuis aux lettres et un seul aux commandes.

Il faut souligner qu'il est difficile de trouver des méthodes comparables à UniGlyph sur tous les points de vue car cette méthode vient notamment pour remédier aux défauts de celles existantes en utilisant un concept totalement nouveau.

Pour pouvoir évaluer ces claviers, nous en avons codé des copies (conformes du niveau fonctionnel) en C#. Concernant Four-Key EdgeWrite, nous avons implémenté l'algorithme de temporisation pour la segmentation décrit par Wobbrock et al., [2006] en utilisant un facteur F de 1,6 qui correspond à la moyenne entre la valeur minimale de 1,2 proposée pour le niveau « expert » et la valeur maximale de 2,0 proposée pour le niveau « novice » par Jacob Wobbrock et ses collègues.

Les trois claviers ont été déployés sur un Pocket PC équipé d'un processeur de 624 MHz, 64Mo de mémoire d'applications et, 256Mo de mémoire de données et d'un écran de 3,7".

Cinq étudiants volontaires, une femme et quatre hommes, âgés entre 22 et 25 ans ont participé à l'expérimentation. Tous les participants sont droitiers (ils tiennent le PDA dans la main gauche et le stylet de la droite), ils n'avaient jamais utilisé un assistant numérique avant cette expérience et n'avaient aucune connaissance au préalable des trois méthodes de saisie à évaluer.

il ne faut pas hurler avec les loups
on parle mal quand on ne veut rien dire
tout usage finit par se changer en abus
la poule a pondu un œuf au poulailler
une baleine a fait un saut spectaculaire
il est toujours content de ses cadeaux
la vache regarde passer les trains

Tableau III-9. Exemples de phrases du corpus de l'évaluation.

Le corpus utilisé pour l'évaluation contient 105 phrases simples composées uniquement de lettres non accentuées (voir exemples dans le Tableau III-9, la liste de phrases complètes est disponible dans l'annexe 6). La longueur des phrases est comprise entre 30 et 40 caractères, espaces compris. La longueur moyenne d'une phrase est de

35,8 caractères. Ce corpus a été créé manuellement et les phrases ont été choisies de façon à ne représenter aucune ambiguïté ou difficulté orthographique et de manière à respecter la longueur moyenne des mots et les fréquences des lettres dans la langue française. Nous avons aussi choisi d'éviter les mots accentués pour assurer une comparabilité avec les méthodes conçues pour une saisie en anglais.

D'autres chercheurs ont déjà mis à disposition un corpus de phrases simples pour être utilisé dans les travaux d'évaluation des entrées de texte [MacKenzie et Soukoreff, 2003]. Il a été effectivement utilisé dans plusieurs travaux comme [Költringer et Grechenig, 2004 ; Wigdor et Balakrishnan, 2004 ; Agarwal et Simpson, 2005] mais nous ne pouvons pas les utiliser dans notre contexte car les participants sont francophones. La différence de la langue peut ralentir la tâche de recopie et causer des erreurs non liées à l'utilisation des claviers étudiés.

L'évaluation s'est déroulée sur cinq sessions. L'intervalle entre les sessions est de 24 à 48 heures (selon la disponibilité des participants). À chaque session, les participants sont invités à recopier, en utilisant chacun des trois claviers, un ensemble de cinq phrases sélectionnées aléatoirement parmi les 105 phrases du corpus. Ces phrases sont présentées en haut de l'écran une par une. L'utilisateur doit entrer un retour à la ligne à la fin de chaque phrase.

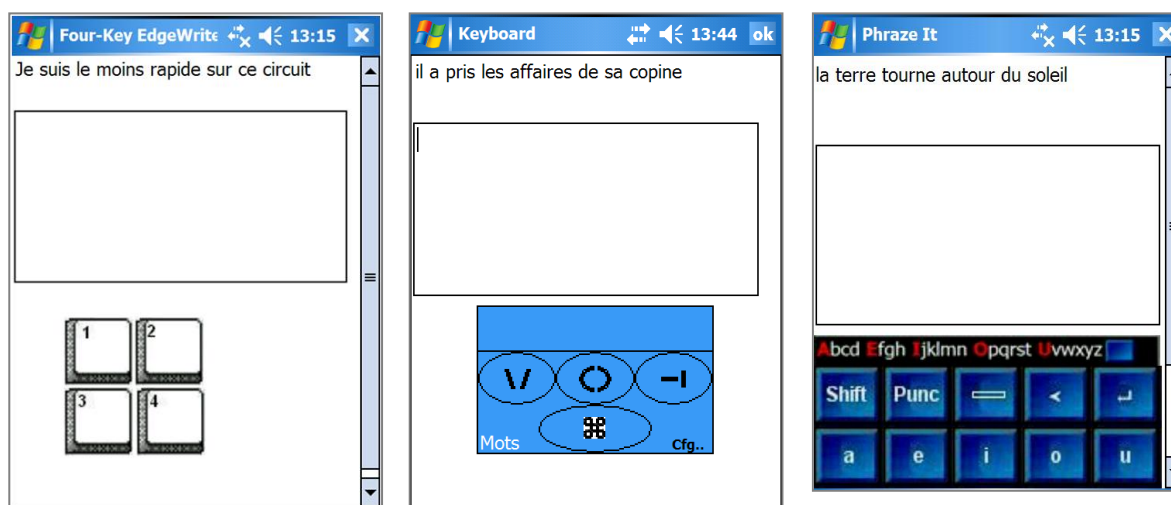


Figure III-24. Prototypes de l'évaluation.

L'instruction principale donnée aux participants au début de l'expérience est de recopier les phrases le plus rapidement possible en corrigeant au maximum les erreurs commises lors de la saisie. La correction de toutes les erreurs n'est pas obligatoire. Ce protocole nous permet d'avoir des conditions très proches d'une saisie naturelle. Il est adopté par beaucoup de chercheurs pour l'évaluation des méthodes de saisie de texte [Soukoreff et MacKenzie, 2003 ; Soukoreff et MacKenzie, 2004 ; Gong et Tarasewich, 2005 ; MacKenzie et al., 2006 ; Yatani and Truong, 2007]. Lors de la saisie, toutes les actions effectuées sur chacun des claviers sont étiquetées temporellement et enregistrées dans un fichier journal (log). En plus de la journalisation, les sessions ont été filmées

mais nous n'avons pas pu analyser ces enregistrements à cause du temps que nécessite une telle tâche.

Avant de commencer la première session, les participants sont invités à lire une fiche qui présente l'évaluation et explique son déroulement (voir annexe 7). À chaque session, chaque participant utilise les trois claviers dans un ordre aléatoirement imposé. Pour chaque clavier, une fiche décrivant le principe de la saisie et éventuellement la table des caractères (cas d'UniGlyph et de 4-Key EdgeWrite) est fournie au testeur (voir annexes 8, 9 et 10). Il doit lire la fiche et, en même temps, essayer le clavier sur le PDA pour mieux comprendre le principe de la méthode. La durée maximale de familiarisation est fixée à 4 minutes pour rester inférieure à la durée d'une session. Durant les trois premières sessions, les participants peuvent s'appuyer sur les fiches d'aide lors de la recopie des phrases. Avec UniGlyph, le feedback sur les touches (Figure III-19) est activé durant les deux premières sessions. Il faut noter aussi qu'avec UniGlyph, l'option de complétion des mots est désactivée durant toute l'expérience.

4.2.2.2. Résultats et discussions

a. La vitesse de saisie

À partir des traces obtenues à la fin de l'évaluation, nous avons pu déduire les performances atteintes pour chaque méthode. Les performances en vitesse de saisie (WPM) sont représentées par les graphes suivants.

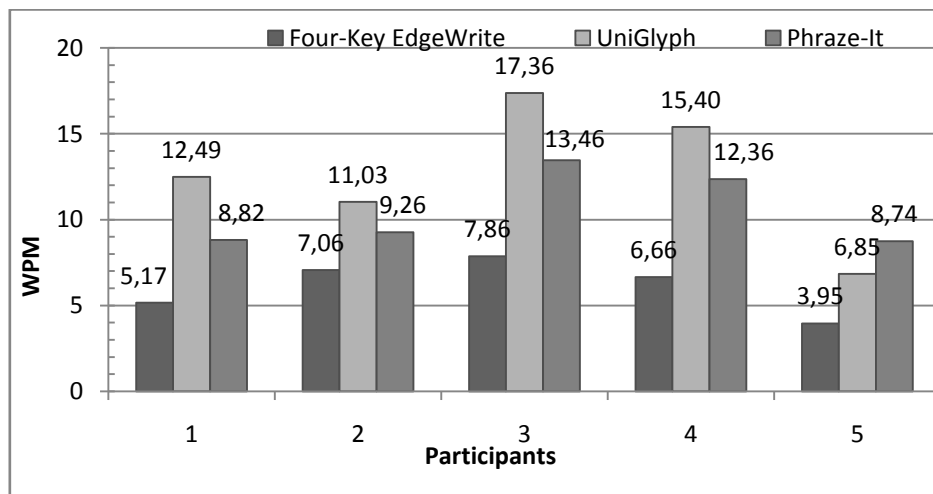


Figure III-25. Nombre maximal de mots saisis par minute (WPM) par participant.

La Figure III-25 montre la vitesse de saisie maximale (en mot par minute) atteinte par chaque participant pour chacun des trois claviers. On peut remarquer que quatre sujets ont pu obtenir la meilleure performance avec UniGlyph. Le cinquième obtient des résultats plus faibles que les autres pour tous les claviers. Ce sujet avait plus de difficultés à utiliser un PDA. Pour UniGlyph, cela s'est traduit par des erreurs de sélection du mot désiré sur la ZAS et une ressaisie complète des mots. Le sujet 3 était le plus performant. Il a atteint une vitesse de 17,36 WPM à la cinquième session en utilisant UniGlyph contre des vitesses maximales de 13,46 et 7,86 respectivement avec Phrase-It et Four-Key EdgeWrite. La moyenne du WPM de tous les utilisateurs à la

cinquième session est de 12,56 pour UniGlyph, 9,36 pour Phraze-It et 5,67 pour Four-Key EdgeWrite.

Le résultat obtenu avec UniGlyph corrobore les estimations faites dans l'étude théorique qui prend en compte le temps de réaction (19,56 WPM). Ce résultat ne représente cependant qu'une évaluation du niveau novice. La Figure III-26 montre bien que la courbe de vitesse moyenne de saisie pour UniGlyph est loin d'avoir atteint son maximum à la cinquième session. Une évaluation séparée avec un utilisateur expert (personne travaillant sur le projet) montre qu'on peut dépasser les 40 mots par minute.

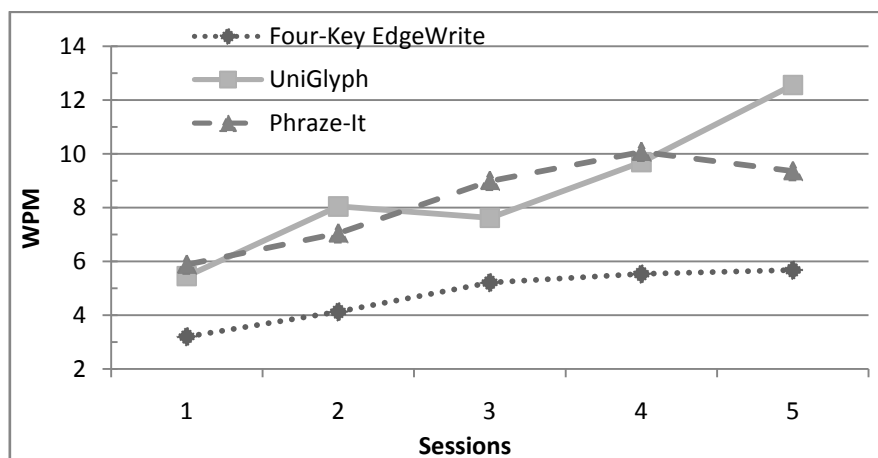


Figure III-26. Nombre moyen de mots saisis par minute (WPM) par session.

On peut remarquer aussi sur la Figure III-26 qu'à la troisième session, la vitesse moyenne a chuté puis qu'elle a remontée à la quatrième et cinquième session. Cela est dû à la désactivation du feedback sur les touches (affichage des lettres autour des primitives) et montre bien que les sujets s'appuyaient sur le feedback aux premières sessions.

Dans cette expérience, nous avons pu calculer aussi le temps que prend l'action du défilement de la ZAS (scrolling) pour trouver le mot désiré. Ce temps correspond à environ 5% du temps de saisie à la cinquième session. Les quatre premiers participants (participants 1, 2, 3 et 4 sur la Figure III-25) passent 3,16% du temps à parcourir la ZAS, le cinquième y passe 12,85% du temps à cause des nombreuses erreurs de sélection. Cette difficulté rencontrée par l'un des cinq sujets nous montre que le mécanisme de défilement par geste horizontal n'est pas aussi facile pour tous les utilisateurs. Des améliorations peuvent être apportées à la ZAS en réduisant sa sensibilité par exemple ou en exploitant la molette physique ou les boutons « + » et « - » disponibles sur la plupart des PDA et sur certains téléphones portables. Cette solution physique permettra de diminuer les erreurs de sélection dues à la difficulté de l'utilisation d'un écran tactile par certains utilisateurs. Elle permettra aussi l'utilisation du clavier UniGlyph par des personnes atteintes de déficiences motrices mineures (tremblements par exemple).

b. L'effort moteur

Nous pouvons également comparer les méthodes en termes de nombre de clics par caractère. Théoriquement, UniGlyph permet une saisie à environ 1 KSPC (KeyStrokes

Per Character) comme nous l'avons expliqué précédemment, Phrase-It à un peu moins de 2 KSPC (vu que l'espace est entré en un seul clic), et Four-Key EdgeWrite entre 3 et 5 KSPC en moyenne selon les représentations des lettres choisies par l'utilisateur.

Dans le graphe de la Figure III-27, les performances en KSPC des trois claviers sont calculées en divisant le nombre total de clics effectués dans chaque session (incluant les clics effectués pour la suppression et la correction des erreurs) par la longueur du texte saisi selon l'équation suivante :

$$KSPC = \frac{|InputStream|}{|TranscribedText|} \quad (1)$$

En plus de l'indication sur l'effort moteur que donne le KSPC théorique, le KSPC obtenue par voie expérimentale donne un aperçu sur le taux d'erreurs [MacKenzie and Tanaka-Ishii, 2007]¹ et par conséquent sur le niveau d'apprentissage et d'adoption de la méthode par l'utilisateur.

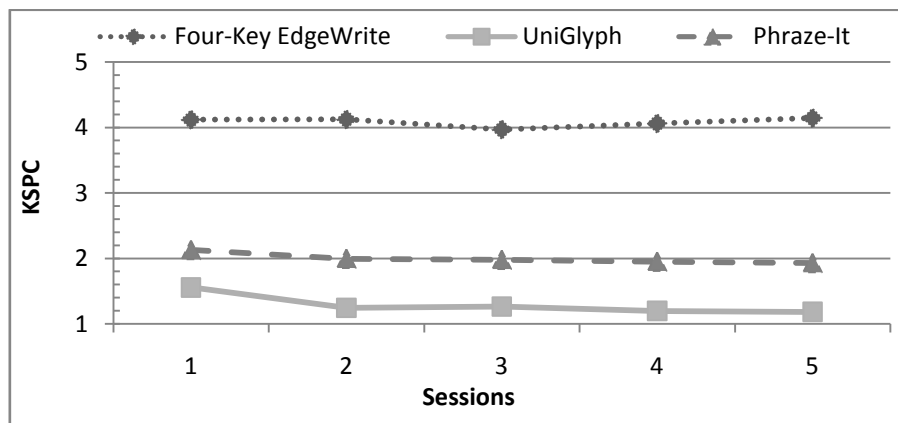


Figure III-27. Nombre moyen d'appuis par caractère (KSPC) par session.

La courbe du KSPC moyen pour UniGlyph converge rapidement vers 1 sans l'atteindre. Cela montre une présence d'erreurs résiduelles même à la cinquième session. Le meilleur KSPC enregistré par un participant est de 1,04 pour UniGlyph, 1,85 pour Phrase-It et 3,38 pour Four-Key EdgeWrite. Les moyennes pour tous les utilisateurs à la cinquième session sont 1,18 pour UniGlyph, 1,93 pour Phrase-It et 4,14 pour Four-Key EdgeWrite.

c. Le taux d'erreurs

Une autre mesure de performance que nous avons pu obtenir est le taux d'erreurs MSD (Minimum String Distance). Nous rappelons que cette mesure prend en compte uniquement le texte final saisi en le comparant au texte à saisir et donne donc un aperçu sur la quantité d'erreurs que l'utilisateur n'a pas corrigé lors de la saisie [Soukoreff et MacKenzie 2001].

¹ Pages: 52-53.

La Figure III-28 montre que le taux d'erreurs moyen pour tous les participants sur les cinq sessions de l'expérience. On peut déduire qu'ils produisent un texte comportant moins d'erreurs avec UniGlyph qu'avec les deux autres méthodes. Cela peut être grâce à la saisie de niveau mot que propose UniGlyph. En effet, dans une saisie lettre par lettre, quand l'utilisateur fait une saisie erronée et poursuit avec une ou plusieurs lettres correctes, il peut considérer que ce n'est pas une « grande » erreur puisque le texte reste compréhensible et il peut continuer la saisie sans la corriger. Dans une saisie de niveau mot par contre, quand l'utilisateur fait une erreur de saisie, cela signifie que le mot entier est erroné et l'erreur est perçue comme « grande » et elle est corrigée systématiquement.

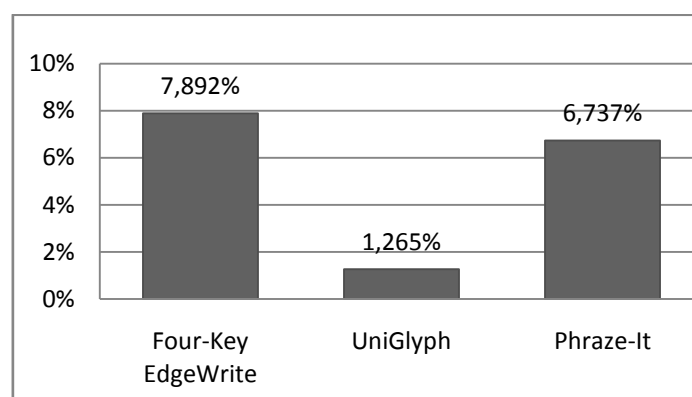


Figure III-28. Taux d'erreurs moyen obtenu avec chacune des trois méthodes.

d. La satisfaction de l'utilisateur

Dans le domaine de l'entrée de texte, les métriques de performances constituent de bons indicateurs sur l'efficacité et la facilité d'utilisation de la méthode évaluée mais ces performances ne sont pas les seules garanties du « succès » d'une méthode de saisie de texte. Un facteur non moins important est la satisfaction de l'utilisateur. Un utilisateur satisfait ne rejette pas facilement le système satisfaisant. Pour cela, nous avons demandé aux sujets de remplir, pour chacune des trois méthodes évaluées, un questionnaire sur une échelle de 5 niveaux (voir Tableau III-10). Cette échelle est communément utilisée dans les questionnaires dans le domaine de la psychologie. Elle est connue sous le nom d'échelle de Likert [Likert et Rensis, 1932].

		Pas du tout d'accord				Totalement d'accord
1	Je trouve que ce clavier permet de saisir facilement du texte	☐	☐	☐	☐	☐
2	Je trouve que ce clavier permet de saisir rapidement du texte	☐	☐	☐	☐	☐
3	Je trouve que le codage des lettres est facile à mémoriser	☐	☐	☐	☐	☐
4	Je trouve que ce clavier permet de corriger	☐	☐	☐	☐	☐

	facilement les erreurs					
5	Je trouve que ce clavier est original	☐	☐	☐	☐	☐
6	J'aimerais bien pouvoir utiliser ce type de clavier sur un assistant numérique	☐	☐	☐	☐	☐

Tableau III-10. Le formulaire d'évaluation de la satisfaction des utilisateurs.

Ce genre d'évaluation subjective en addition aux mesures de performances empiriques est souvent utilisé dans le domaine de la saisie de texte [Cockburn et Siresena, 2003 ; Amar et al., 2003 ; Martin, 2005 ; Cox et al., 2007]. Certains travaux utilisent d'autres échelles de notation dans leurs questionnaires. Karlson et Bederson [2007] par exemple, utilisent une échelle de 7 niveaux.

Les résultats obtenus à la fin de l'expérience sont transcrits dans le Tableau III-11.

Question	Méthodes	Sessions				
		1	2	3	4	5
1	UniGlyph	2,2	3	3,6	3,4	4
	Phraze-It	2,8	3,4	3,4	3,8	3,8
	4-Key EdgeWrite	2,8	3,6	3,4	3	3
2	UniGlyph	2,4	3,8	3,6	3,2	3,8
	Phraze-It	2,8	3,4	3,2	3,8	3,8
	4-Key EdgeWrite	2,4	3,2	3,8	3,2	3
3	UniGlyph	2,2	3,2	3	2,6	3,2
	Phraze-It	3,2	3,6	3	3,2	3,4
	4-Key EdgeWrite	2,6	3	3,4	3,6	3,6
4	UniGlyph	2,4	2,2	2,6	3,2	3,6
	Phraze-It	4,2	3,8	3,6	3,8	4
	4-Key EdgeWrite	3,2	3,2	3	2,4	2,6
5	UniGlyph	4,2	4,2	4	4,2	3,8
	Phraze-It	3,4	3,4	3,2	3,4	3,2
	4-Key EdgeWrite	3,8	4	3,6	4	3,4
6	UniGlyph	2,2	3,2	3,2	3,2	3,4
	Phraze-It	2,8	3,2	2,8	3,4	3
	4-Key EdgeWrite	2,4	3,4	3,4	3	2,8

Tableau III-11. Résultats des questionnaires de satisfaction.

Les moyennes des notes sont calculées à partir des formulaires remplis par les participants en considérant que la réponse « pas du tout d'accord correspond à une note de 1 et que la réponse « totalement d'accord » correspond à une note de 5.

À partir de ce tableau, on ne remarque pas d'évolutions claires de l'avis des utilisateurs sur chacune des méthodes mais on peut en extraire des courbes plus expressives. La Figure III-29 montre les moyennes des notes attribuées par tous les utilisateurs à chacune des questions et pour chacune des méthodes à la cinquième session.

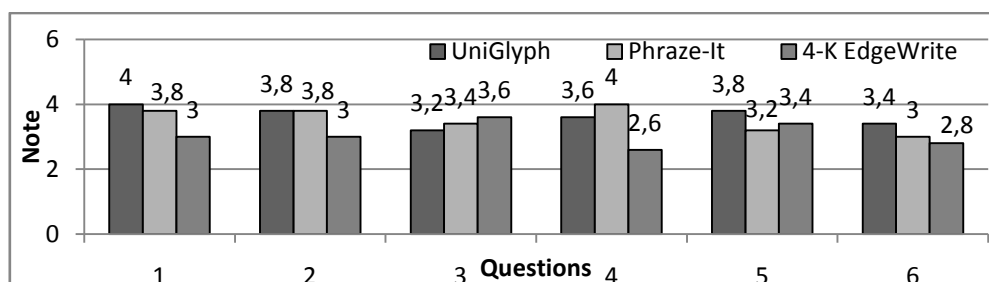


Figure III-29. Avis des participants sur les trois méthodes à l'issue de la cinquième session.

À partir du graphe de la Figure III-29, on peut remarquer que les utilisateurs trouvent que la saisie avec UniGlyph est la plus facile des trois méthodes (question 1) et qu'elle est originale comme solution de saisie (question 5). Ils trouvent aussi qu'UniGlyph permet de saisir rapidement du texte (question 2) mais ils la mettent au même niveau que Phrase-It. Dans les résultats empiriques décrits précédemment, UniGlyph affiche de meilleures vitesses de saisie (moyenne et maximale) que Phrase-It. La raison pour laquelle les utilisateurs mettent ces deux méthodes au même rang peut être due au grand écart de vitesse qu'elles ont avec la troisième méthode (4-Key EdgeWrite).

Pour la question sur la facilité de mémorisation du codage des caractères (question 3), l'écart entre les trois méthodes est très serré. UniGlyph est considérée comme la plus difficile et 4-Key EdgeWrite comme la plus facile à mémoriser. Enfin, les participants considèrent que Phrase-It est la méthode qui permet de corriger le plus facilement les erreurs (question 4) car la suppression correspond à un simple appui de touche contre deux appuis avec UniGlyph et 4-Key EdgeWrite. Cependant, la correction des erreurs avec UniGlyph est considérée beaucoup plus facile qu'avec 4-Key EdgeWrite. Cela peut être dû à la possibilité de supprimer un mot entier avec UniGlyph quoique la comparaison soit difficilement faisable car cette dernière permet une saisie de niveau mot et donc il est logique de permettre une suppression de niveau mot.

Un autre graphique qu'on a pu extraire du tableau précédent concerne les moyennes de notes obtenues à chaque session pour chaque méthode sur l'ensemble des questions (voir Figure III-30).

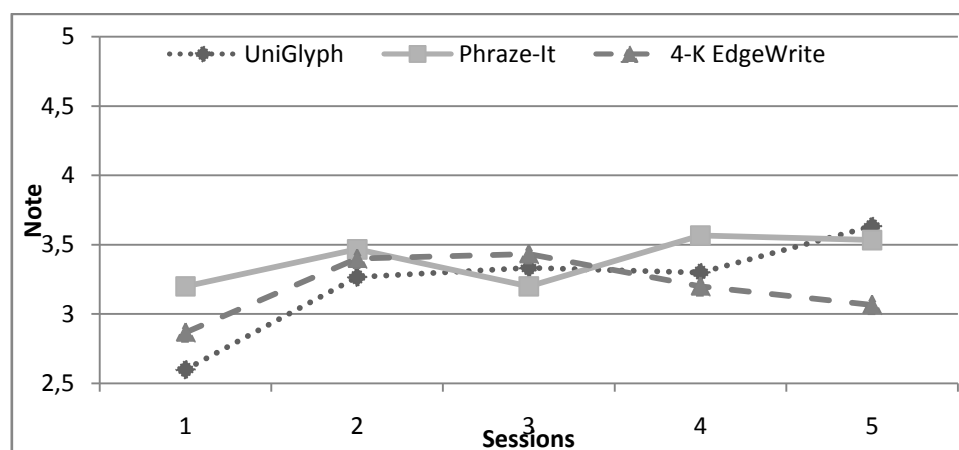


Figure III-30. Évolution de la notation de chaque méthode au long de l'expérience.

De ce graphique, la courbe la plus facile à interpréter est celle d'UniGlyph. L'appréciation des utilisateurs monte continuellement entre la première et la dernière session. La note moyenne passe de la dernière position à la première en dépassant les deux autres méthodes. Pour Phrase-It, les variations ne sont pas très significatives. On peut dire que l'utilisateur reste sur le même point de vue ou que son appréciation augmente très légèrement. Enfin, pour 4-Key EdgeWrite, l'appréciation par les utilisateurs augmente au départ car ils comprennent facilement le principe d'analogie. Cette appréciation redescend aussitôt car ils s'aperçoivent que la saisie n'est pas rapide et qu'elle est fatigante.

5. Autres implémentations

Comme nous l'avons signalé précédemment, le clavier UniGlyph permet d'utiliser les touches physiques du pavé directionnel du PDA. Cela permet d'économiser encore plus d'espace sur l'écran mais aussi de réduire les erreurs qui peuvent être dues à la difficulté d'interaction avec un écran tactile à cause d'un handicap moteur léger (tremblements par exemple). Cette fonctionnalité peut être étendue aussi au mécanisme de défilement et de sélection des mots. Le défilement par geste horizontal sur l'écran peut être remplacé par un défilement à molette physique cliquable comme on trouve sur beaucoup de téléphones mobiles, et de plus en plus sur les PDA (voir Figure III-31). Cela permettra aussi d'implémenter UniGlyph sur d'autres dispositifs mobiles qui ne disposent pas d'écran tactile comme les téléphones mobiles classiques.

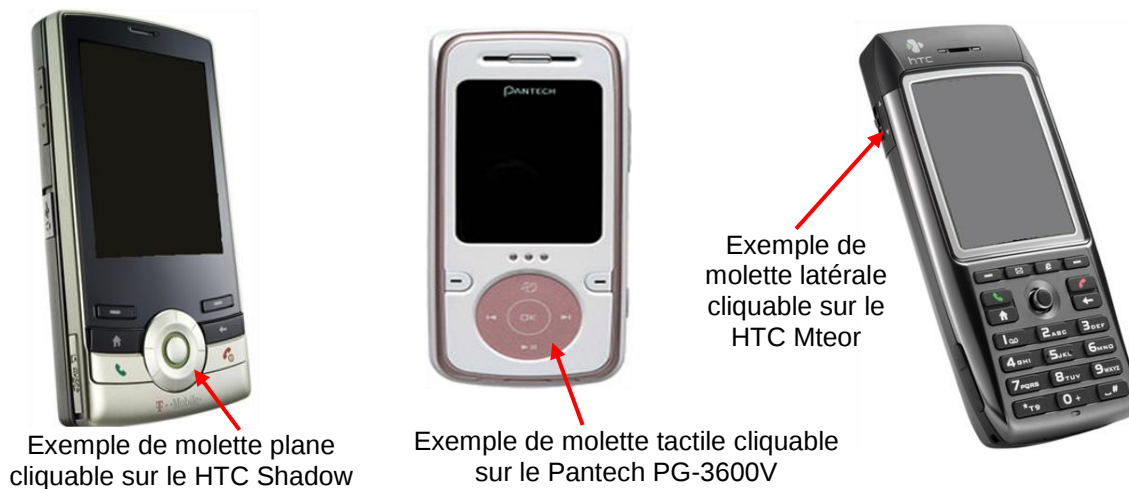


Figure III-31. Exemples de PDA, de téléphone mobile et de Smartphone équipés de différents types de molettes qui peuvent être exploitées par UniGlyph.

6. Conclusion

UniGlyph offre une solution de saisie de texte avec un KSPC très proche de 1 sur un clavier minimal de trois touches de primitives. La méthode UniGlyph est basée sur des techniques communes en saisie de texte sur dispositifs mobiles pour être indépendante du dispositif. Son implémentation est possible sur un grand nombre d'appareils mobiles sous forme de claviers physiques ou logiciels, aussi bien avec une interaction au doigt qu'au stylet.

Dans ce mémoire, nous avons présenté une implémentation d'UniGlyph qui utilise un dictionnaire de mots en français mais UniGlyph peut supporter d'autres langues basées sur les caractères romains. Pour l'anglais¹ l'étude statistique préliminaire a donné une moyenne de 2,31 candidats par séquence de primitives avec la même table de caractères. D'autres langues avec d'autres alphabets comme l'arabe par exemple peuvent bien être représentées par les même six primitives d'UniGlyph car ces dernières représentent tous les traits de base qu'on peut trouver dans un dessin, à savoir, les traits inclinés, les traits arrondis (sans considération de l'orientation) et les traits droits.

L'évaluation préliminaire en ligne ainsi que l'évaluation contrôlée montrent bien que l'apprentissage de la nouvelle méthode ne nécessite pas une longue pratique. La prédiction de la vitesse de saisie avec le clavier logiciel UniGlyph selon les lois de Fitts et de Hick-Hyman donne une vitesse de saisie de 19,56 mots par minute.

L'évaluation empirique avec des utilisateurs novices confirme que l'apprentissage de la méthode est rapide. Après cinq sessions de recopie de cinq phrases courtes par session, la vitesse de saisie maximale obtenue est de 17,36 WPM. Ce résultat est très proche de la vitesse prédite théoriquement. UniGlyph offre un compromis entre les diverses qualités requises par une méthode de saisie pour dispositifs mobiles que sont : la rapidité, les petites dimensions, la grande taille des touches, la facilité de l'apprentissage et la versatilité.

Un autre point fort de la méthode de saisie UniGlyph est qu'elle est indépendante du dispositif. Dans l'implémentation décrite dans ce chapitre, le clavier UniGlyph intègre une option qui permet d'utiliser des touches physiques pour la saisie. Cette idée peut être améliorée et adaptée pour les utilisateurs avec des déficiences motrices qui ne leur permettent pas d'utiliser une interface tactile. Mais pour les personnes avec un handicap lourd, cette solution est totalement inadaptée. Pour répondre à ce besoin, et en partant du constat de l'état de l'art qui souligne le manque de solutions pour ce type d'utilisateurs, nous avons décidé d'implémenter une variante d'UniGlyph adaptée aux handicaps moteurs lourds.

¹ ANC (American national corpus) environ 240 000 entrée. Disponible à : <http://americannationalcorpus.org/>

CHAPITRE IV

HANDIGLYPH : L'ADAPTATION D'UNIGLYPH POUR HANDICAP MOTEUR LOURD

Résumé

Ce chapitre décrit une autre implémentation de la méthode UniGlyph pour les personnes ayant un handicap moteur lourd. Cette nouvelle implémentation est baptisée « HandiGlyph ». Elle s'inspire des systèmes de communication palliative tout en assurant une meilleure autonomie grâce à l'utilisation d'un dispositif mobile et une vitesse de saisie maximale due au faible nombre de touches de la méthode de base, « UniGlyph ».

1. Introduction

Comme nous l'avons souligné dans le chapitre I, il existe quelques méthodes de saisie sur dispositifs mobiles conçues pour les utilisateurs présentant des déficiences visuelles ou des déficiences motrices légères. Pour les personnes ayant un handicap moteur lourd, le domaine est encore à explorer. À part quelques tentatives timides par des personnes qui sont confrontées directement au besoin de l'accessibilité des personnes avec handicaps lourds aux dispositifs d'infocommunication mobiles (notamment KeCom, développé au centre de rééducation CMRRF de Kerpape), on ne trouve que des systèmes de communication palliative AAC (Augmentative and Alternative Communication) dédiés aux ordinateurs de bureau et aux ordinateurs portables. On peut même se poser la question « à quoi sert de donner l'accessibilité aux dispositifs de communication mobiles pour des personnes à mobilité réduite ? ». Cette question nous a été posée dans une conférence par un chercheur dans le domaine de l'AAC et nous y répondons ici pour enlever l'ambiguïté que posent les mots « dispositif mobile » et « mobilité réduite ».

Ces dispositifs sont de plus en plus performants et intègrent de plus en plus de fonctionnalités qui nous facilitent la vie et qui sont accessibles en permanence : agenda, notes, contacts, messagerie, navigation, synthèse vocale, etc. Ces fonctionnalités sont en général les mêmes que celles dont peut avoir besoin la personne handicapée sur un ordinateur traditionnel sauf que si elles sont accessibles sur un petit dispositif, l'utilisateur handicapé n'aura pas à aller au bureau pour en tirer profit. Il n'aura pas non plus à transporter un ordinateur portable (lourd et encombrant) sur son fauteuil roulant et il gagnera en termes d'autonomie en énergie grâce à la différence de poids et éventuellement à la différence de consommation d'énergie entre l'ordinateur portable et le dispositif mobile.

Le deuxième élément de réponse concerne le gain en temps qu'on peut avoir grâce à un dispositif mobile. Au lieu d'aller chercher son ordinateur de bureau, la personne handicapée peut avoir en permanence son assistant numérique sur son fauteuil roulant. Même dans le cas où la personne peut avoir en permanence son ordinateur portable sur le fauteuil roulant, il lui faut attendre quelques minutes pour le démarrer ou l'initialiser alors que le mode de fonctionnement des dispositifs mobiles ne nécessite pas de redémarrage ni d'hibernation entre utilisations.

Ces deux points sont, pour nous, les principales justifications de l'utilité de tels dispositifs pour les personnes avec des déficiences motrices sévères mais d'autres arguments peuvent être non moins importants. Par exemple, les fonctionnalités de ces dispositifs informatiques mobiles peuvent encore être étendues pour satisfaire les besoins de la personne handicapée (système de télé-alerte en cas de personne isolée, contrôle de l'environnement par communication infrarouge/Bluetooth, communication et transfert de données sans fil avec les autres systèmes informatiques de l'environnement, etc.).

Pour que ces petits dispositifs soient utilisables par cette catégorie d'utilisateurs, il faut bien sûr y apporter quelques adaptations (système de navigation adapté, menus adaptés....). L'un des éléments clés de l'utilisation d'un dispositif mobile est son système d'entrée de texte comme nous l'avons souligné précédemment. Dans ce chapitre nous présentons une implémentation de la méthode UniGlyph adaptée à une utilisation par les personnes avec un handicap moteur lourd. Cette nouvelle implémentation est baptisée « HandiGlyph ».

2. Principes et interfaces

2.1. Principes

HandiGlyph est une méthode de saisie de texte dédiée aux personnes ayant des déficiences motrices sévères, par exemple, infirmités motrices cérébrales (IMC), sclérose latérale amyotrophique (SLA), syndrome de l'enfermement (LIS) ou quelques formes de tétraplégies et spécialement aux personnes qui ne peuvent effectuer que des mouvements très limités comme le mouvement de la main ou du bras, le mouvement de la tête, de la langue ou des paupières. Un simple contacteur (aussi appelé bouton poussoir ou switch en anglais) permet de détecter l'action de l'utilisateur et de la traduire en un événement de clic ou d'appui de bouton pour le dispositif mobile.

Le clavier HandiGlyph est basé sur les mêmes principes d'UniGlyph détaillés précédemment en plus de quelques techniques communes dans le domaine de la communication palliative, à savoir : le balayage des touches et l'utilisation d'un simple capteur à deux états (contacteur). Le balayage consiste à mettre successivement en surbrillance les éléments de l'interface sur l'écran et à sélectionner l'élément désiré en agissant sur le contacteur au moment où cet élément est mis en surbrillance [Harris et Vanderheiden, 1980].

2.2. Interface du clavier

L'interface de HandiGlyph est composée de trois touches ambiguës de deux primitives chacune, d'une touche de commande (comme le clavier UniGlyph pour personnes valides) et de deux Zones d'Affichage et de Sélection (Voir Figure IV-1).

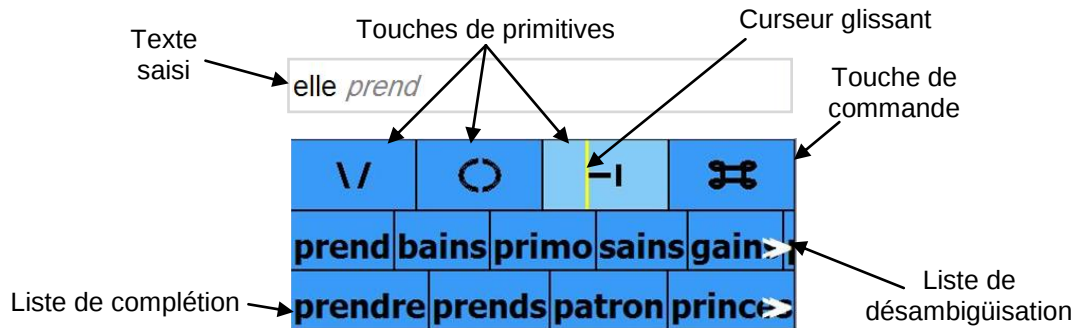


Figure IV-1. Le clavier HandiGlyph est composé de 4 touches et de deux zones d'affichage et de sélection.

Les touches de primitives et la touche de commande sont balayées séquentiellement de gauche à droite. À chaque moment, l'utilisateur peut appuyer sur le contacteur (bouton-poussoir) pour sélectionner la touche courante. Nous avons ajouté un curseur glissant (timing line) qui parcourt la touche active pour indiquer le temps restant avant que la touche suivante soit activée. Ce type de feedback temporel est très utile pour certains utilisateurs. Il a été utilisé dans Sibylle [Wandmacher et al., 2008]. Dans la suite, nous utiliserons l'expression « cliquer sur la touche X » pour exprimer que l'utilisateur appuie sur le contacteur lorsque la touche X est active.

À chaque appui sur une touche de primitives, le clavier génère deux listes de mots, une liste de désambiguïsation qui comprend les mots qui correspondent exactement à la séquence des primitives entrée et une liste de complétion qui correspond aux mots les plus probables qui commencent par la séquence de primitives entrée. Ces deux listes sont affichées sur deux zones d'affichage et de sélection.

Dans l'implémentation précédente d'UniGlyph pour personnes valides, nous n'avons pas utilisé une liste de complétion séparée car la vitesse de saisie était beaucoup plus importante. L'ajout d'une ZAS supplémentaire impliquait l'ajout d'éléments supplémentaires dans l'interface et ainsi l'augmentation du temps de réaction selon la loi de Hick-Hyman et la réduction de la vitesse de la saisie [Goodenough-Trepagnier et al., 1986]. Dans le cas d'HandiGlyph, la vitesse de saisie est limitée par la vitesse de balayage. L'utilisateur a largement le temps de chercher le mot désiré sur les deux listes.

Pour valider un mot, l'utilisateur doit sélectionner la ZAS concernée en effectuant un clic long ou un clic très long sur l'une des touches de primitives pour accéder respectivement à la liste de désambiguïsation ou à la liste de complétion. Les différentes longueurs (périodes) de clics peuvent être définies selon les préférences et les capacités de l'utilisateur. Deux types de retours (feedbacks) sont utilisés pour permettre à l'utilisateur de différencier les types de clics :

- Un retour sonore qui consiste à émettre un son pour marquer le passage du clic simple au clic long, du clic long au clic très long et du clic très long au clic simple.
- Un retour visuel qui consiste à changer la couleur et l'épaisseur de la touche (ou l'élément) cliquée.

Si l'utilisateur ne peut pas effectuer les clics longs, le cycle de balayage peut être étendu pour inclure les deux zones d'affichage (les différentes configurations sont expliquées en section 2.3). Dans ce cas le curseur glisse horizontalement de gauche à droite sur les trois touches de primitives et sur la touche de commande puis verticalement de haut en bas sur les deux ZAS. Pour en sélectionner une, l'utilisateur effectue un simple clic quand elle est activée.

Une fois la ZAS sélectionnée, son contenu est balayé de gauche à droite mot par mot. Quand le dernier mot visible est atteint, le défilement automatique de la liste (scroll) affiche la suite des mots prédits. Pour valider le mot désiré, l'utilisateur doit effectuer un simple clic quand ce dernier est sélectionné. À tout instant, l'utilisateur peut reprendre le cycle de balayage initial sur les quatre touches de la première ligne, sans choisir un mot, en effectuant un clic long. Dans ce cas aussi, si l'utilisateur ne peut effectuer des longs clics, l'alternative est de laisser défiler tous les éléments de la liste. Le curseur de balayage reprend son cycle normal à partir de la première touche de primitives.

En plus de la saisie des mots (mode MOT), HandiGlyph offre les mêmes possibilités et aussi les mêmes modes que le clavier UniGlyph décrit dans le chapitre précédent, avec de légères différences :

- Le mode de commande est activé lorsque l'utilisateur clique sur la touche de commande (appuis simple, long ou très long). Il permet d'effectuer les suppressions, les espacements et les retours à la ligne. En fonction du délai du clic sur la touche de commande, la première touche permet de supprimer un caractère, un mot ou d'effacer toute la ligne (voir Figure IV-2). Le mode de commande donne aussi accès aux autres modes de saisie. Les abréviations des noms de ces modes sont affichées sur la zone d'affichage au milieu du clavier.

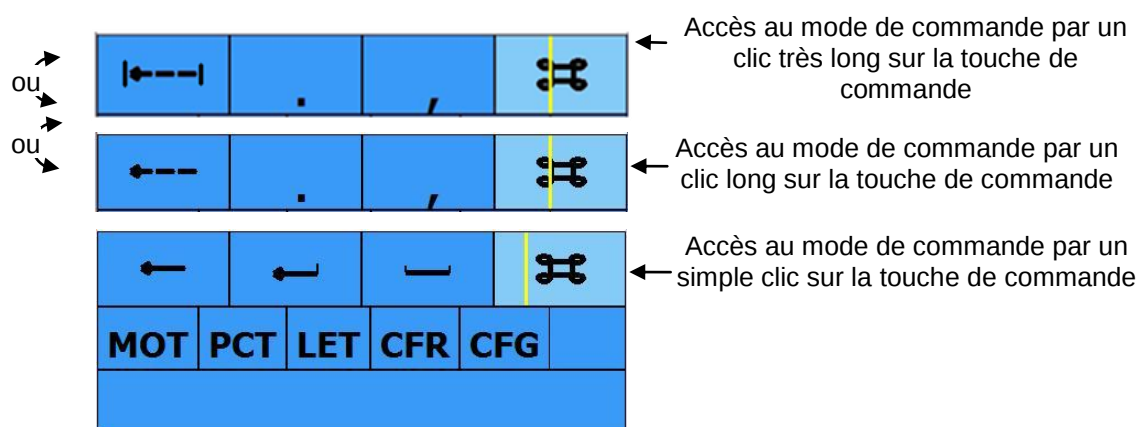


Figure IV-2. Dans le mode de commande, les touches de primitives deviennent des touches de fonctions et la ZAS du milieu permet d'accéder aux autres modes.

- Le mode de saisie des lettres est accessible par le raccourci « LET » (le même que dans UniGlyph, voir chapitre III).
- Le mode d'entrée des marques de ponctuation est accessible par le raccourci « PCT » (le même que dans UniGlyph).
- Le mode de saisie des chiffres est accessible par le raccourci « CFR » (le même que dans UniGlyph).

2.3. Interface de configuration

En mode de commande, la zone d'affichage du milieu affiche un cinquième raccourci (CFG) qui permet d'ouvrir une fenêtre de configuration. Cette fenêtre est divisée en deux onglets : un onglet principal qui permet à l'utilisateur handicapé de personnaliser lui-même son système sans avoir besoin de l'aide d'une personne assistante et un deuxième onglet qui est réservé aux configurations les moins fréquemment utilisées et qui sont donc effectuées par une personne assistante comme dans les systèmes d'AAC classiques. Dans cette section, nous nous consacrons à la partie des configurations manipulables par la personne handicapée elle-même.

Le principe du balayage est utilisé dans cette fenêtre pour permettre à la personne handicapée d'agir sur les différents objets de l'interface de configuration. Seuls des objets simples qui peuvent être manipulés en un simple clic sont utilisés dans cette interface (cases à cocher, boutons et palettes de couleurs). Un focus parcourt séquentiellement les différents éléments de l'interface et l'utilisateur peut effectuer un simple clic pour cocher/décocher une case à cocher, appuyer sur un bouton (+/-) ou sélectionner une palette (voir Figure IV-3).

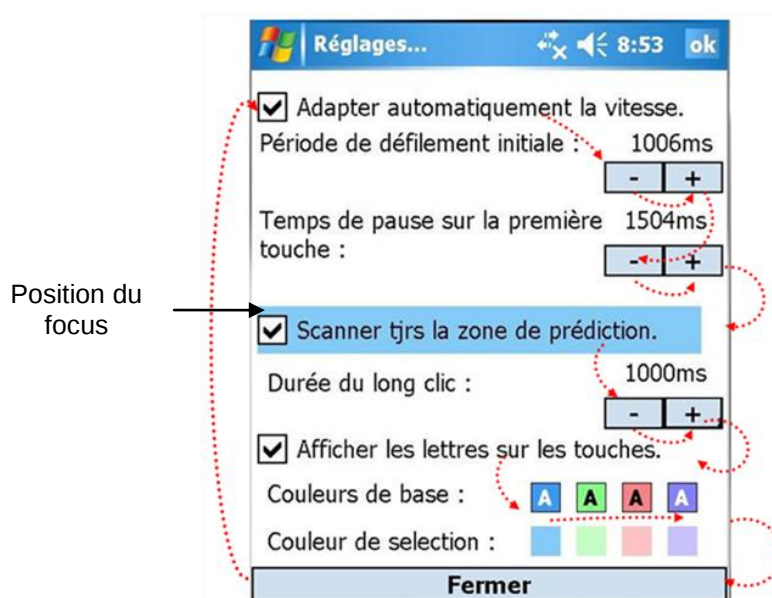


Figure IV-3. Dans la fenêtre de configuration de HandiGlyph, les différentes options sont balayées en continue.

Cette interface permet de manipuler les éléments de configurations suivants :

a. Adaptation automatique de la vitesse de balayage

Toutes les personnes handicapées n'ont pas les mêmes capacités motrices et ils ne peuvent pas interagir avec un système automatisé à la même vitesse. C'est pour cela que

nous avons implémenté un module d'ajustement automatique de la fréquence F du balayage qui recalcule, en cours de la saisie, la fréquence du balayage en se basant sur des données collectées en temps réel. La technique d'adaptation utilisée est inspirée et adaptée d'un algorithme simple de Lesh et al., [2000]. Cet algorithme a été adapté, implémenté et évalué dans certains systèmes de communication palliative comme EDiTH [Ghedira et al., 2006] par exemple.

Sur chaque 20 clics effectués par l'utilisateur (entrée de primitive, sélection d'un mot, etc.), la période du balayage est multipliée par 1,05 si l'une des conditions suivantes est vérifiée :

- Trois suppressions ou plus,
- Trois cycles de balayage sans action de la part de l'utilisateur.

Pour la dernière condition, si ces trois cycles sont successifs, ils ne sont pas pris en compte dans cet algorithme car on considère qu'ils n'ont pas été manqués mais que l'utilisateur n'est tout simplement pas en train de saisir du texte. Les pauses et la réflexion en cours de la saisie sont courantes même pour les personnes valides et ne doivent pas entraîner un ralentissement du balayage.

La réalisation de l'une des deux conditions précédentes signifie que l'utilisateur rate le clic sur l'objet sélectionné car il clique tardivement et, dans ce cas, il est obligé d'effectuer une correction, ou dans le deuxième cas, il n'a pas le temps de cliquer et donc il est obligé d'attendre le prochain cycle pour effectuer la sélection. Cela indique que la fréquence F du balayage est élevée et donc doit être réduite en multipliant la période T par 1,05.

Dans les autres cas, quand aucune des conditions précédentes n'est réalisée, on calcule le rapport entre le temps de réaction de l'utilisateur pour appuyer sur une touche et le temps de sélection de cette touche (période T du balayage). Si la moyenne de ce rapport sur les 20 actions étudiées est inférieure à 65%, cela signifie que l'utilisateur a encore 35% du temps de marge et donc que la période du balayage est trop élevée. Elle est factorisée par 0,95 dans ce cas pour la réduire. Si aucune de ces conditions n'est réalisée, la fréquence du balayage reste inchangée.

L'utilisateur peut aussi désactiver la case à cocher correspondante à cette option pour garder la même fréquence de balayage tout au long de l'interaction.

b. Ajustement manuel de la période du balayage

Par défaut, à la première utilisation de HandiGlyph, la période du balayage est réglée à 2 secondes. L'utilisateur peut l'ajuster manuellement au lieu d'attendre la convergence de l'ajustement automatique. Les deux boutons « + » et « - » (voir Figure IV-3) permettent d'augmenter cette période ou de la réduire de 10 ms à chaque appui simple. Le bouton appuyé reste sélectionné pour d'autres éventuels appuis successifs jusqu'à ce que la période T du balayage soit écoulée après le dernier appui. Pour une action plus rapide et, éventuellement, moins fatigante, l'utilisateur a la possibilité de garder le bouton appuyé

jusqu'à ce que la valeur désirée soit atteinte. Cela équivaut à une succession très rapide d'appuis simples.

c. Ajustement du temps de sélection de la première touche

La première touche de primitives « $\backslash/$ » et le premier mot dans les zones d'affichage sont souvent sélectionnés juste après que l'utilisateur effectue une action (quand une touche de primitives est cliquée, le cycle de balayage recommence par la première touche « $\backslash/$ » et quand l'utilisateur sélectionne une ZAS pour choisir un mot, le balayage commence par le premier mot affiché). Pour éviter que l'utilisateur rate le clic sur ce premier élément balayé, un temps de perception est nécessaire en plus du temps normal de sélection des autres éléments, en l'occurrence la période du balayage.

L'utilisateur peut donc définir manuellement le temps de sélection du premier élément balayé selon ses préférences et capacités. Ce temps ne peut pas être inférieur à la période normale du balayage définie par l'option précédente. Le système enregistre le rapport entre ce temps et la période du balayage normale pour la recalculer quand cette dernière est modifiée soit automatiquement ou manuellement.

d. Configuration du cycle de balayage

Par défaut, le focus de balayage passe seulement sur les trois touches de primitives et la touche de commande pour économiser le temps de défilement sur les touches. L'utilisateur effectue un appui long ou un appui très long, comme nous l'avons expliqué précédemment, pour accéder à l'une des ZAS. Dans certains cas, les longs clics peuvent être fatigants, voir impossibles pour certains utilisateurs. L'option « scanner toujours les zones de prédiction » permet d'étendre le cycle du balayage aux zones d'affichage en créant une sorte de balayage mixte : séquentiel sur les quarts touches de la première ligne et ligne/colonne sur les deux autres lignes (les deux ZAS). Cette technique nous permet de faire un compromis entre la vitesse de saisie et le nombre d'appuis par caractère. Un balayage ligne/colonne classique signifie une saisie à 2 KSPC (un clic pour valider la ligne et un autre pour sélectionner la touche désirée). Un balayage séquentiel simple sur tous les éléments de l'interface (les touches puis les mots affichés sur les deux ZAS un par un) ne peut pas être utilisé car cela signifie une augmentation trop importante du temps nécessaire de saisie. En outre, dans le cas d'une erreur, l'utilisateur sera obligé d'attendre le défilement de tous les mots des deux listes pour pouvoir effectuer la correction. La combinaison des deux solutions nous permet de conserver la saisie à un appui par caractère et la sélection des mots à deux appuis. Le cycle du balayage est prolongé de $2 \times T$. Au lieu de $4 \times T$ dans le cycle court, on a $6 \times T$ dans le cycle de balayage prolongé (avec T : la période de balayage).

e. Ajustement du délai du clic long

Par défaut, le délai du clic long est d'une seconde (affichée en millisecondes : 1000 ms). L'utilisateur peut changer cette durée en utilisant les boutons « + » et « - » comme dans les options précédentes. Le clic « très long » est toujours le double du clic long.

f. Affichage des lettres sur les touches

Comme avec le clavier UniGlyph, cette option fournit une aide à l'utilisateur lors de la phase de familiarisation pour faciliter l'apprentissage et éviter le rejet immédiat de la méthode.

g. Choix des couleurs

L'interface propose quatre palettes prédéfinies de couleurs d'arrière plan, couleur du texte et couleur de sélection. Le focus de balayage défile de gauche à droite sur les quatre palettes et l'utilisateur peut en choisir une par un simple clic.

En plus de ces options, l'interface contient un bouton pour fermer la fenêtre de configuration. Ce bouton est inclus dans le balayage de l'interface pour remplacer la petite croix de fermeture des fenêtres Windows qui ne peut pas être utilisée par une personne handicapée.

3. Évaluation d'HandiGlyph avec une personne handicapée

En collaboration avec le Centre Mutualiste de Rééducation et de Réadaptation Fonctionnelles (CMRRF) de Kerpape, nous avons organisé une évaluation de HandiGlyph avec une personne handicapée pour avoir un retour utilisateur sur son utilisabilité et son acceptabilité par le public concerné. Le seul participant à cette évaluation que nous pouvons qualifier de « test utilisateur » est un volontaire de 26 ans avec le syndrome d'enfermement (Locked-In Syndrome). Il faut souligner la difficulté de trouver des testeurs volontaires qui acceptent de se déplacer au centre de Kerpape avec leur assistant de vie et de prendre le temps de tester un nouveau système, autre que celui qu'ils utilisent quotidiennement (notamment sur ordinateur de bureau).

3.1. Matériel et application expérimentale

Dans cette expérience, nous avons utilisé un PDA Phone Qtek 9100 sous Windows Mobile™ 5.0 équipé d'un processeur de 200 MHz, une mémoire de stockage de 64 Mo, une mémoire programme de 128 Mo et un écran tactile de 2,8". Nous avons utilisé aussi une souris Bluetooth pour Pocket PC (Stowaway® de ThinkOutside) pour déclencher les événements de clics dans l'interface. La souris ne joue pas le rôle d'un dispositif de pointage dans notre mécanisme. Elle est reliée à un contacteur (bouton poussoir) et sert uniquement à transmettre les actions de ce dernier au PDA car nous ne disposons pas de contacteur qui puisse être connecté directement au PDA par câble ou autre. Cette solution permet aussi de connecter tout type de contacteur ou de capteur utilisé par le testeur (voir Figure IV-4). Ce matériel a été fourni par le laboratoire d'électronique du CMRRF de Kerpape.

L'autre alternative qui a été implémentée est de pouvoir utiliser n'importe quel bouton du pavé directionnel du Pocket PC ou d'agir directement aux doigts sur l'écran tactile pour effectuer les clics. Dans ce dernier cas, l'endroit du contact du doigt avec l'écran n'a aucune signification, seul l'évènement du clic est considéré.



Figure IV-4. La souris Bluetooth équipée d'un connecteur pour brancher le contacteur utilisé par le testeur.

Une application qui implémente le clavier HandiGlyph a été développée spécialement pour cette expérience. Elle couvre tout l'écran pour détecter les clics sur l'ensemble de sa surface. Elle affiche aussi l'heure et l'état de la batterie car elle couvre la barre des tâches de Windows (la barre horizontale en haut de l'écran qui contient entre autre le menu « démarrer »). Le module de désambiguïsation et de prédiction est le même que celui décrit précédemment dans l'évaluation d'UniGlyph (basé sur un modèle de langage uni-gramme).

3.2. Protocole de l'expérience

Le participant à cette expérience est bien familiarisé avec les dispositifs informatiques. Il a déjà testé plusieurs systèmes d'AAC comme Sibylle [Wandmacher et al., 2007] et il utilise le système KeCom développé à Kerpape pour la saisie de texte sur PDA (voir chapitre I). Pour ces raisons, et puisque nous avons déjà évalué le niveau novice sur UniGlyph, nous avons décidé de laisser l'utilisateur se familiariser avec HandiGlyph avant de procéder à notre expérience. Nous lui avons expliqué oralement les principes de la nouvelle méthode et lui avons fourni une carte descriptive qui contient la carte des caractères et la règle de saisie (voir annexe 11). Nous l'avons laissé ensuite se familiariser avec le clavier HandiGlyph pendant une durée de deux mois durant lesquels il l'a souvent utilisé comme moyen de communication écrite. Il saisissait ses phrases et faisait signe à son interlocuteur pour les lire sur l'écran du PDA. La restriction des possibilités offertes par notre application de test ne l'a pas encouragé à l'utiliser très fréquemment (pas de synthèse vocale, ni d'envoi de courriel ni de SMS alors que KeCom lui offre toutes ces possibilités). Toutefois, il a eu le temps de se familiariser avec HandiGlyph.

Après la période de familiarisation, une séance d'évaluation a été organisée au sein du laboratoire d'électronique du CMRRF et supervisée par un ingénieur du laboratoire. Le participant a été invité à recopier des phrases courtes extraites aléatoirement de la liste de phrases utilisées précédemment dans l'évaluation d'UniGlyph.

Durant l'expérience, le participant est dans son fauteuil roulant, le PDA installé dans son socle en face de lui sur le plateau du fauteuil. La souris Bluetooth et le contacteur sont aussi disposés sur le plateau. La phrase à recopier s'affiche en haut de l'écran du PDA, une zone de saisie au milieu et le clavier HandiGlyph en bas de l'écran (voir Figure

IV-5). Le participant est invité à recopier la phrase affichée et à la valider par un retour à la ligne pour afficher la phrase suivante. La saisie de cinq phrases constitue une session, le participant fait une courte pause entre les sessions.

Durant cette expérience, la fonction d'adaptation automatique de la vitesse de balayage est active avec une période de balayage initiale de 810 ms. Le cycle de balayage inclut seulement les touches de primitives et la touche de commandes. La transition aux zones d'affichages se fait par appui long/très long.

La fonction d'adaptation automatique de la vitesse de balayage est activée durant cette expérience, avec une période de balayage initiale de 810 ms. Le cycle de balayage inclut seulement les touches de primitives et la touche de commandes. La transition aux zones d'affichages se fait par appui long/très long.

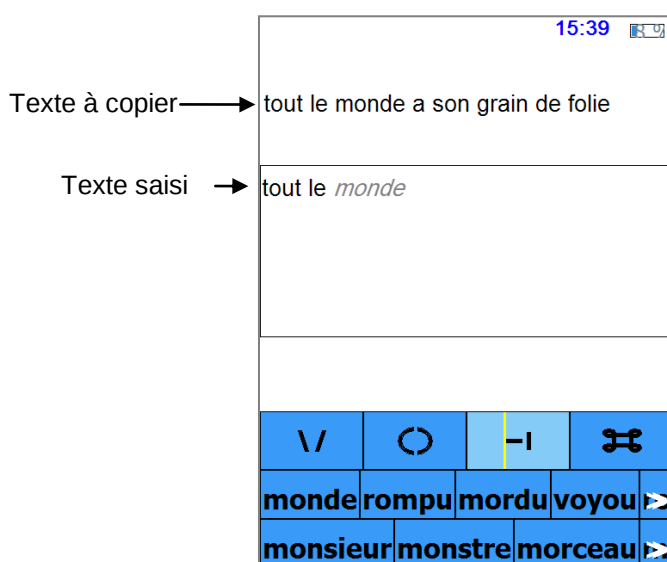


Figure IV-5. Copie d'écran du prototype expérimental.

3.3. Résultats et discussions

Lors de l'expérience, tous les événements ont été étiquetés temporellement et enregistrés dans des fichiers de journalisation. Au total, cinq sessions de cinq phrases chacune ont été effectuées. En se basant sur la journalisation de l'expérience, nous avons pu déduire les éléments suivants.

3.3.1. L'effort

Le KSPC permet d'évaluer le travail moteur nécessité par une méthode de saisie pour entrer du texte. Comme dans les expériences précédentes, nous calculons le KSPC en divisant le nombre d'appuis effectués par l'utilisateur par la longueur du texte transcrit. Les appuis simples, longs et très longs sont tous considérés comme des « keystrokes ». Nous n'avons pas trouvé de travaux antérieurs qui séparent ces différents types d'appuis pour le calcul du KSPC. La longueur des appuis influencera plutôt la vitesse de saisie.

La Figure IV-6 montre le KSPC obtenu à chaque session de l'expérience. Il varie entre 1,61 et 1,42. La moyenne du KSPC sur les cinq sessions est de 1,5. Cela signifie une

présence d'erreurs lors de la saisie entraînant un effort moteur supplémentaire pour les corriger.

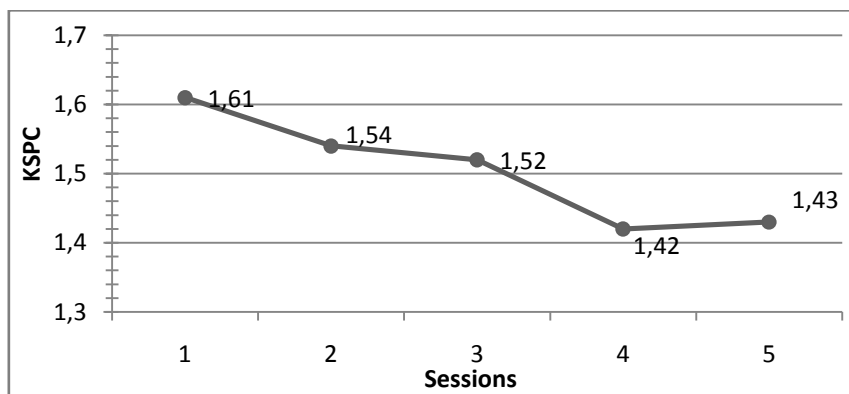


Figure IV-6. Nombre d'appuis par caractère sur cinq sessions.

Nous avons aussi pu constater qu'à plusieurs reprises, l'utilisateur entre un mot qui ne correspond pas exactement à celui qu'il doit saisir mais qui constitue sa première partie. Il bascule ensuite en mode de saisie des lettres pour entrer le suffixe manquant. Par exemple, dans le cas de la saisie du mot « profondeur », après avoir effectué sept appuis de touches de primitives, le mot « profond » s'affiche sur la liste de désambiguïsation. L'utilisateur le valide, puis il bascule en mode de saisie des lettres pour entrer les lettres 'e', 'u' et 'r'. Cela est très coûteux en termes de nombre d'appuis effectués pour saisir le mot. Dans le cas cité ici, au lieu de faire 12 appuis, l'utilisateur en fait 24 pour saisir le mot « profondeur ». Ces 24 appuis correspondent à la saisie du mot « profond » (9), la suppression de l'espace inséré automatiquement après ce mot (2), le passage en mode de saisie des lettres (3), la saisie de la séquence de lettres « eur » (5), le retour en mode de saisie des mots (3) et enfin la saisie d'un espace (2).

La raison de la répétition de cette erreur qui n'a pas été observée lors des évaluations d'UniGlyph est que le participant utilise quotidiennement d'autres systèmes d'AAC qui permettent de prédire la fin d'un mot déjà saisi en partie (e.g. Sybille). Cette fonctionnalité n'est pas intégrée dans la version actuelle d'HandiGlyph. Son implémentation peut être très intéressante en termes d'effort moteur fourni lors de la saisie. La liste de complétion actuelle permet uniquement de compléter la saisie en cours. En d'autres termes, elle affiche seulement les mots qui commencent par la séquence de primitives entrées et non pas ceux qui commencent par le mot déjà validé. Il faut noter que cette option n'est pas prise en considération dans les méthodes prédictives à saisie ambiguë les plus largement utilisées comme T9®.

L'autre constat qu'on a pu faire, est que l'utilisateur supprime à plusieurs reprises les mots validés par erreur, lettre par lettre, au lieu d'utiliser la fonctionnalité de suppression de mot. Cela donne deux appuis par caractère supprimé au lieu de deux appuis pour tout le mot (un appui long sur la touche de commande et un appui simple sur la touche de suppression).

3.3.2. Le taux d'erreurs

La Figure IV-7 montre le taux d'erreurs résiduelles dans le texte final transcrit. Ce taux d'erreurs est obtenu par comparaison du texte à copier au texte transcrit selon l'algorithme du MSD (Minimum String Distance) décrit précédemment. On peut noter qu'il y a très peu d'erreurs non corrigées. Dans la deuxième session, l'utilisateur n'a pas supprimé un mot erroné, mais il a placé le mot correct juste après. À la cinquième session, le texte résultant contient un espacement supplémentaire (double espace au lieu d'un seul). Comme nous l'avons déjà souligné dans le chapitre précédent, le principe de la saisie par mot permet d'avoir un texte avec très peu d'erreurs.

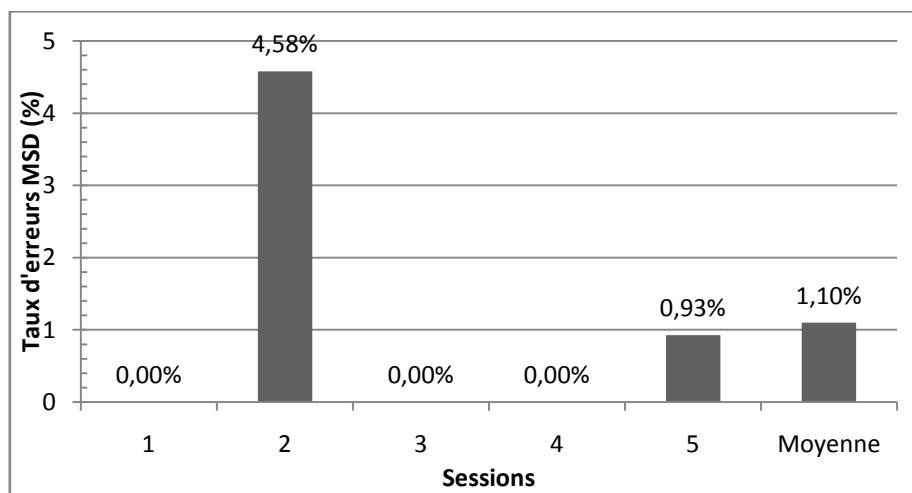


Figure IV-7. Les taux d'erreurs résidentielles dans le texte transcrit.

3.3.3. Les données relatives au temps

La vitesse de saisie n'est pas très significative dans ce contexte. Elle dépend beaucoup des capacités motrices et aussi du degré du handicap de l'utilisateur. Dans cette expérience, la vitesse de saisie moyenne observée sur les cinq sessions est de 2,37 mots par minute (WPM). Cette vitesse a évolué de 1,22 WPM à la première session à 3,18 WPM à la cinquième session. Il faut dire que la vitesse du balayage a évolué aussi d'une moyenne de 754 ms à la première session à 118 ms à la cinquième comme le montre la Figure IV-8.

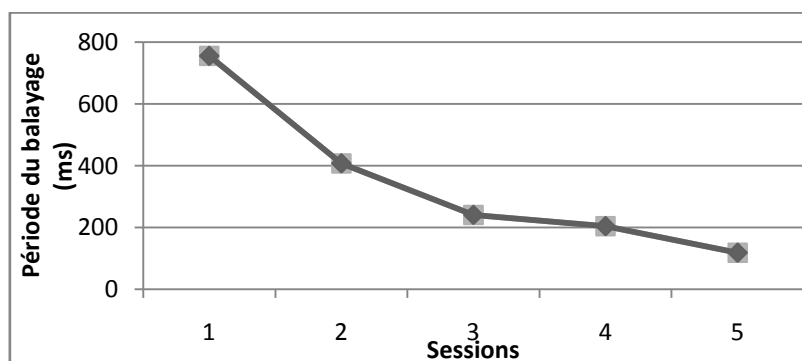


Figure IV-8. Évolution de la vitesse du balayage.

D'autres variables relatives au temps et non moins importantes ont pu être extraites à partir des fichiers de journalisation de l'expérience menée à Kerpape. Le temps passé

sur chacune des deux zones d'affichage est détaillé par la Figure IV-9. Le temps passé sur une ZAS est délimité par la fin du clic long ou très long qui permet d'y accéder et le moment où le curseur de défilement quitte cette ZAS soit parce que l'utilisateur a effectué un clic pour sélectionner un mot, soit parce qu'il a effectué un clic long pour revenir au cycle de défilement initial sans sélectionner de mot, soit parce qu'il a laissé défiler la liste de mots jusqu'à la fin sans faire aucune action (dans ce cas aussi le balayage reprend son cycle initial sur les quatre touches de la première ligne).

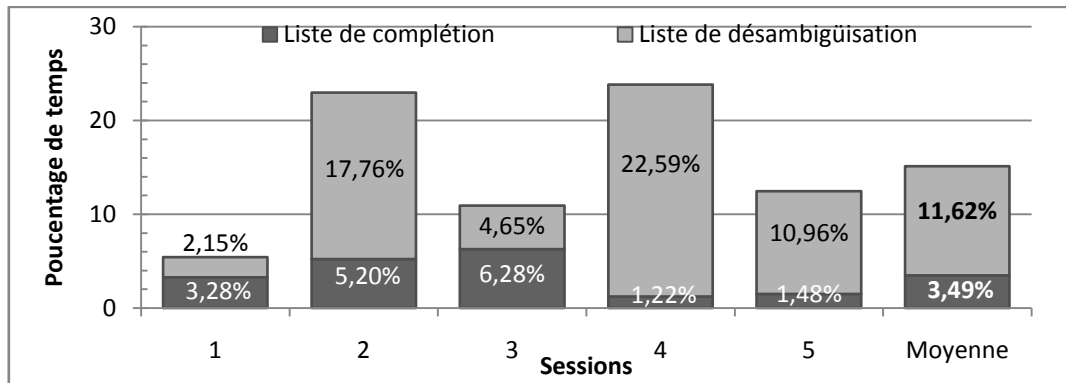


Figure IV-9. Pourcentages du temps passé sur les ZAS par rapport au temps total de la tâche de saisie.

Pour faciliter la lecture de la Figure IV-9, voici un exemple : dans la première session, le participant passe 2,15% du temps de saisie sur la ZAS de la liste de désambiguïsation et 3,28% de ce temps sur la ZAS qui affiche la liste de complétion. Le temps passé sur les deux ZAS à la première session est donc de $2,15\% + 3,28\% = 5,43\%$ du temps de saisie.

On peut remarquer que l'utilisateur passe plus de temps sur la zone d'affichage et de sélection qui correspond à la liste de désambiguïsation que sur celle de la liste de complétion. Cela n'est pas dû au nombre de mots sélectionnés à partir de chacune. La Figure IV-10 montre que le nombre de mots sélectionnés à partir de chacune des deux listes est presque le même.

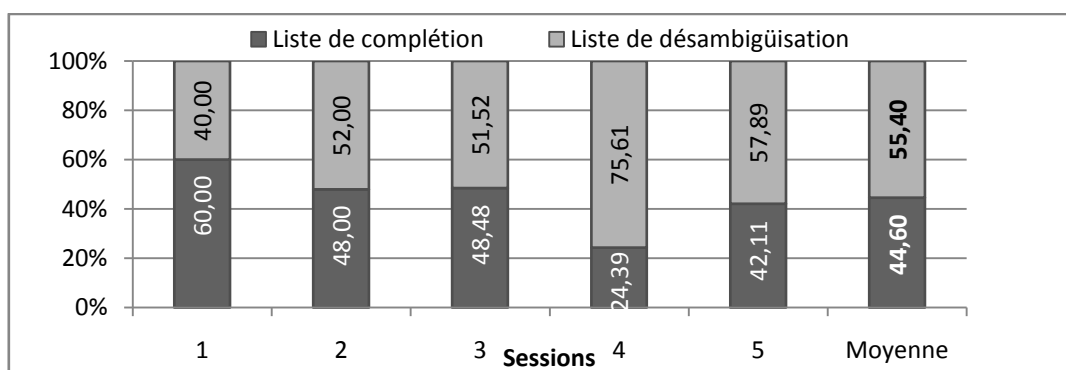


Figure IV-10. Pourcentages des mots trouvés dans chaque liste.

En effet, l'utilisateur passe à la liste de complétion seulement quand il voit que le mot désiré est présent. Cela n'est pas le cas pour la première ZAS car elle affiche la liste des mots qui correspondent exactement à la séquence de primitives entrée : quand l'utilisateur pense avoir entré la bonne séquence de primitives, il passe au balayage de

cette liste même si le mot désiré n'est pas affiché car ce dernier peut être dans la suite de la liste et pourra donc être atteint par balayage et défilement automatique. Cependant, si la séquence entrée n'est pas correcte, l'utilisateur ne s'en rend compte qu'après le défilement d'un certain nombre de mots, voire toute la liste sans trouver le mot désiré. Dans ce cas il modifie la séquence de primitives entrées et corrige l'erreur avant de repasser par la ZAS pour sélectionner le mot à saisir.

Même quand la séquence de primitives est correcte, si le mot désiré n'est pas parmi les plus fréquents de la liste, il n'est pas affiché parmi les premiers. L'utilisateur doit donc attendre le défilement jusqu'à ce que ce mot soit affiché et atteint par le curseur de balayage.

3.3.4. La satisfaction de l'utilisateur

Comme nous l'avons fait lors de l'évaluation d'UniGlyph et comme il est d'usage dans beaucoup de travaux d'évaluation des entrées de texte, nous avons procédé à une évaluation subjective en plus de la déduction des métriques empiriques. Le participant a été invité, après l'expérience, à compléter le formulaire du Tableau IV-1 en notant chaque question sur une échelle à 5 niveaux. Il faut noter que le formulaire a été transmis au participant par courriel et qu'il l'a complété chez lui avec ses propres moyens (client courriel adapté sur PC) sans la présence d'aucune personne en relation avec le projet.

	Pas du tout d'accord				Tout à fait d'accord
Je trouve que ce clavier permet de saisir facilement du texte	☐	☐	☐	☒	☐
Je trouve que ce clavier permet de saisir rapidement du texte	☐	☐	☐	☐	☒
Je trouve que le codage des lettres est facile à mémoriser	☐	☐	☒	☐	☐
Je trouve que ce clavier permet de corriger facilement les erreurs	☐	☐	☒	☐	☐
Je trouve que ce clavier est original	☐	☐	☐	☒	☐
J'aimerais bien pouvoir utiliser ce type de clavier avec une synthèse vocale pour la communication	☐	☐	☐	☐	☒

Tableau IV-1. Le formulaire d'évaluation du niveau de satisfaction de l'utilisateur

Les réponses transcrites dans le Tableau IV-1 par le participant montrent essentiellement un grand intérêt à HandiGlyph. Il est convaincu que cette méthode lui permet de saisir très rapidement du texte avec peu d'effort.

Il nous a aussi exprimé le souhait d'avoir une version plus libre hors du cadre de l'évaluation contrôlée pour pouvoir l'utiliser au quotidien. Une application qui intègre

HandiGlyph avec quelques fonctionnalités dont une synthèse vocale et un client de messagerie est en cours de développement. Elle pourrait permettre de faire une évaluation dans des conditions naturelles sur longue période.

4. Conclusion

Nous avons présenté dans ce chapitre une implémentation sur PDA et pour personnes handicapées moteurs de la méthode UniGlyph décrite en début du chapitre précédent. Cette nouvelle implémentation s'appelle HandiGlyph et vise les utilisateurs atteints de handicaps moteurs lourds qui ne leur permettent d'interagir avec le système que par un simple contacteur à capteur binaire.

Les techniques de balayage utilisées dans les systèmes d'aide à la communication traditionnels ont été associées aux principes de la saisie UniGlyph pour construire le clavier HandiGlyph. Le nombre réduit de touches (4 seulement) limite le cycle du balayage et permet d'avoir un affichage statique des touches contrairement aux claviers à balayage qui utilisent un mécanisme de réorganisation dynamique des touches selon la probabilité d'apparition de chaque lettre. Avec HandiGlyph, l'utilisateur peut anticiper facilement l'appui suivant la lettre en cours de saisie. Cela lui permet d'utiliser des vitesses de balayage maximales, et atteindre ainsi des vitesses de saisie maximales.

Le test utilisateur effectué au CMRRF de Kerpape donne une vitesse de saisie très encourageante de 3,18 WPM et le testeur est très intéressé par cette nouvelle façon de saisir du texte et il souhaite bien pouvoir l'utiliser comme outil d'aide à la communication.

CONCLUSIONS ET PERSPECTIVES

Dans ce mémoire de thèse, nous avons présenté une méthode de saisie de texte indépendante du dispositif et adaptable aux besoins spécifiques des utilisateurs. Notre contribution se situe autant dans le domaine des nouveaux modes d'interaction avec les dispositifs électroniques mobiles que dans le domaine de l'aide à la communication pour personnes handicapées.

La méthode proposée (UniGlyph) repose sur une saisie analogique par segmentation des caractères en formes de base. Les principes et les objectifs de la méthode ont été déduits à partir d'une large étude des techniques de saisie de texte existantes. Le premier objectif est d'avoir un clavier qui occupe très peu de surface d'affichage dans le cas de solution logicielle et/ou très peu de surface sur le corps matériel du dispositif dans le cas d'une solution physique. Le deuxième objectif est de construire une méthode polyvalente qui peut être utilisée sur différents types de dispositifs et par des utilisateurs valides comme par des utilisateurs handicapés. Pour atteindre ce deuxième but, il faut éviter de concevoir une méthode dépendante de l'interaction au stylet. La solution a été de partir sur une méthode à touches. Dans un contexte tactile, la méthode peut profiter de l'utilisation du geste mais reste implémentable sur les dispositifs qui n'intègrent pas de surfaces tactiles comme les téléphones mobiles classiques par exemple. Un autre critère déterminant est la dimension des touches qui doivent être suffisamment larges pour éviter les erreurs de saisie et permettre une saisie aux doigts même sur un dispositif de très petites dimensions. La taille des touches et le nombre de touches utilisées (et ainsi la distance entre les touches) agissent directement sur la vitesse de saisie selon les lois de Fitts et de Hick-Hyman.

En partant de ces objectifs, des techniques utilisées dans les méthodes de saisie existantes et des travaux réalisés préalablement dans le cadre du projet GLYPH, nous avons sélectionné les meilleures techniques qui peuvent être retenues pour atteindre ces objectifs. Cette combinaison de techniques existantes donne naissance à une méthode de saisie à touches ambiguës de primitives totalement nouvelle qui permet une saisie à un appui de touche par caractère.

Le problème que pouvait poser cette nouvelle approche réside dans l'apprentissage de ce nouveau mode de saisie « codé ». Pour éviter ce problème nous avons étudié minutieusement l'analogie sur laquelle se base notre table de caractères. Nous avons retenu le codage le plus logique pour l'utilisateur et le plus facile à retenir. Dans ce but, nous avons fait une expérience en ligne qui a rassemblé 58 participants de profils différents pour saisir un total de 16 422 caractères. Cette pré-évaluation nous a permis de choisir la meilleure table de caractères pour la nouvelle méthode UniGlyph. La table de caractères choisie repose sur une analogie aux caractères romains en majuscules avec une règle simple et claire pour le choix de la primitive qui représente chaque caractère

sur un clavier de trois touches de deux primitives chacune. L'expérience en ligne a montré que dès les premiers essais les utilisateurs novices arrivent à saisir correctement du texte et sont séduits par la méthode. Après uniquement trois essais de 22 mots chacun, les testeurs ne font plus que 2,07% d'erreurs.

Pour valider notre idée et notre étude « de faisabilité », nous avons implémenté UniGlyph dans un premier temps sur Pocket PC pour utilisateurs valides. Le clavier logiciel UniGlyph est composé des trois touches ambiguës de primitives, d'une touche de commandes et d'une zone d'affichage qui permet de proposer à l'utilisateur la liste des mots correspondants aux appuis qu'il a effectués. La zone d'affichage utilise un principe de zoom de type FishEye qui a montré son utilité dans des applications autre que la saisie de texte pour la visualisation des données sur des dispositifs d'affichage. Dans cette implémentation, nous avons essayé aussi de tirer profit des facultés tactiles de ce genre de dispositifs sans figer l'utilisation de la méthode à cet aspect. Cela permet d'atteindre une large palette d'utilisateurs ayant des capacités différentes d'interaction avec un dispositif à écran tactile. Les commandes gestuelles ont été choisies sur les bases des méthodes existantes en respectant une analogie gestuelle avec la commande à effectuer (e.g., geste vers la gauche pour la suppression, vers la droite pour l'espacement, vers le bas pour le passage à la ligne, vers le haut pour la mise en majuscule).

Nous avons évalué cette implémentation d'UniGlyph avec 5 utilisateurs novices selon un protocole d'expérimentation contrôlée standard de recopie de texte. Les résultats de l'évaluation confirment les prédictions théoriques de la vitesse de saisie. L'un des participants a atteint une vitesse de 17,36 WPM à la cinquième session alors que moyenne a été de 12,56 WPM pour les cinq participants. Ces vitesses de saisie sont très satisfaisantes pour des novices en comparaison avec d'autres méthodes. Dans un niveau expert, l'auteur de cette thèse atteint plus de 40 WPM avec UniGlyph.

L'expérience menée avec les utilisateurs novices nous a révélé cependant que certains utilisateurs peuvent avoir beaucoup de difficulté à utiliser le mécanisme de défilement tactile des mots sur la zone d'affichage. Ce problème peut être réglé de différentes manières : la première consiste à définir plusieurs niveaux de sensibilité tactile comme le font certaines méthodes de saisie (TenGo® par exemple). La deuxième solution consiste à utiliser une molette pour le défilement, sinon de deux boutons physiques sur le dispositif (par exemple les boutons de réglage du son). La troisième alternative est d'exploiter l'accéléromètre intégré à certains dispositifs. D'après l'état de l'art que nous avons effectué au début de cette thèse, la saisie avec les méthodes qui utilisent cette technologie comme TiltText, TiltType ou Unigesture est rapidement fatigante. Cependant, une utilisation plus naturelle de la détection de l'inclinaison peut être réalisée sur le mécanisme d'affichage et de défilement des mots. En effet, on peut se référer à la métaphore d'un objet sur une surface plane dans la nature, avec l'effet de la gravité, quand on incline la surface plane, l'objet se penche vers le côté le plus bas. Cette métaphore peut être utilisée pour le défilement des mots en tant qu'objets sur la surface plane de l'écran. L'interaction apparaîtrait naturelle en suivant un principe physique et pourrait être combinée au défilement tactile en option pour permettre à l'utilisateur de choisir son mode d'interaction préféré.

Un autre point qui peut être amélioré dans le clavier UniGlyph est son prédicteur de mots. En effet, dans nos prototypes, nous avons essayé de voir les limites extrêmes d'un clavier ambigu à trois touches en intégrant un dictionnaire de 41 000 mots. Ce dictionnaire permet, effectivement, de réduire le nombre de passages par le mode de saisie des lettres pour entrer les mots non reconnus, mais en réalité, le vocabulaire utilisé dans la communication humaine est beaucoup moins large [Rosenberg et Tunney, 2008] et encore plus restreint dans le cadre de la communication électronique.

La deuxième implémentation de la méthode UniGlyph sous le nom de « HandiGlyph » est adaptée aux personnes handicapées moteurs lourds. UniGlyph peut être adaptée à différents types de handicaps mais nous avons choisis spécialement ce type de handicap à cause du manque de travaux et de solutions qui visent cette catégorie d'utilisateurs comme nous l'avons souligné dans l'état de l'art. On trouve en effet, beaucoup plus de travaux concernant les déficiences motrices mineures et les déficiences visuelles alors que les dispositifs informatiques mobiles peuvent être aussi (voire plus) utiles aux personnes avec des handicaps moteurs lourds.

HandiGlyph utilise un système de balayage sur les quatre touches d'UniGlyph pour permettre un cycle de balayage court et ainsi une saisie relativement rapide contrairement aux systèmes AAC classiques. La lenteur de la saisie et de la communication pour ces personnes est un élément de frustration quotidienne. Le nombre réduit de touches permet aussi de minimiser la taille du clavier et donne ainsi la possibilité d'avoir une vue plus large sur le texte édité et sur l'application utilisée.

Dans le clavier HandiGlyph, nous avons séparé la liste de désambiguïsation de la liste de complétion pour permettre une économie de saisie et réduire ainsi la fatigue. Nous avons aussi intégré une interface de configuration à balayage pour assurer une grande autonomie de l'utilisateur et réduire la dépendance aux personnes assistantes.

Le système a été testé par une personne avec le syndrome d'enfermement. Les résultats montrent une vitesse de saisie très encourageante de 3,18 WPM. Il faut souligner que cette vitesse ne reflète pas le niveau maximal de performance de la méthode. Le système de complétion peut être modifié de façon à reconnaître et compléter les mots déjà saisis.

Nous pensons qu'une évaluation à long terme dans des conditions naturelles (production de texte) serait la meilleure façon d'évaluer HandiGlyph. Actuellement nous développons une application complète qui intègre HandiGlyph et permet d'envoyer des courriels et des SMS. La zone d'affichage inférieure du clavier reste inexploitée en mode de commande dans nos prototypes actuels. Elle pourrait être utilisée pour intégrer les différentes commandes nécessaires pour ces applications (création courrier, création SMS, envoi, saisie d'adresse électroniques).

Enfin, nous pensons que les claviers UniGlyph et HandiGlyph répondent à un réel besoin en permettant une saisie rapide, non fatigante et non encombrante sur les dispositifs électroniques de petites dimensions.

RÉFÉRENCES

- [Agarwal et Simpson, 2005] Agarwal, A. et Simpson, R. (2005). User modeling for individuals with disabilities: a pilot study of word prediction. In *Proceedings of the 7th international ACM SIGACCESS Conference on Computers and Accessibility* (Baltimore, MD, USA, October 09 - 12, 2005). Assets '05. ACM, New York, NY, 218-219. DOI= <http://doi.acm.org/10.1145/1090785.1090838>
- [Al Faraj et al., 2008a] Al Faraj, K. Vigouroux, N. Mojahid, M. (2008). Seven-Segment Input 7SI: New Stylus-Based Text Entry Method on Mobile Devices. In *Proceedings of the IEEE International Conference on Information and Communication Technologies: from Theory to Applications* (ICTTA 2008), Damascus, Syria, 07-APR-08-11-APR-08, IEEEExplore digital library, 1-6, 2008.
- [Al Faraj et al., 2008b] Al Faraj, K., Mojahid, M., et Vigouroux, N. (2008). GrooveWrite: A Multi-purpose Stylus-Based Text Entry Method. In *Proceedings of the 11th international Conference on Computers Helping People with Special Needs* (Linz, Austria, July 09 - 11, 2008). K. Miesenberger, J. Klaus, W. Zagler, and A. Karshmer, Eds. Lecture Notes In Computer Science, vol. 5105. Springer-Verlag, Berlin, Heidelberg, 1196-1203. DOI= http://dx.doi.org/10.1007/978-3-540-70540-6_180
- [Amar et al., 2003] Amar, R., Dow, S., Gordon, R., Hamid, M.R., et Sellers, C. (2003). Mobile ADVICE: an accessible device for visually impaired capability enhancement. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, April 05 - 10, 2003). CHI '03. ACM, New York, NY, 918-919. DOI= <http://doi.acm.org/10.1145/765891.766069>
- [Baharath et Madhvanath, 2008] Bharath, A. et Madhvanath, S. (2008). FreePad: a novel handwriting-based text input for pen and touch interfaces. In *Proceedings of the 13th international Conference on intelligent User interfaces* (Gran Canaria, Spain, January 13 - 16, 2008). IUI'08. ACM, New York, NY, 297-300. DOI= <http://doi.acm.org/10.1145/1378773.1378814>
- [Bederson, 2000] Bederson, B., (2000). Fisheye Menu Selection, In *Proceedings of ACM User Interface Software and Technology Conference* (San Diego, California, USA, November 5 – 8, 2000) UIST'00, 217-225.
- [Belatar et Poirier, 2007a] Belatar, M. et Poirier, F., (2007), Glyph2PPC: a reduced onscreen keyboard for text input on PDA, In *HuMaN 07 posters*, (Timimoun, Algeria, March 12-14 2007).
- [Belatar et Poirier, 2007b] Belatar M. et Poirier F., (2007), UniGlyph : une méthode universelle pour la saisie de texte sur dispositifs mobiles, In *Proceedings of La conférence de l'association francophone de l'Interaction Homme-Machine IHM'07*, (Paris, France, November 2007) , ACM Press, 111-118.
- [Belatar et Poirier, 2008a] Belatar, M. et Poirier, F., (2008), HandiGlyph : Une méthode de saisie de texte sur dispositifs mobiles pour personnes handicapées, In *Proceedings of La conférence de l'association francophone de l'Interaction Homme-Machine IHM'08* (Metz, France, September 2 – 5, 2008), ACM Press, 81-84.
- [Belatar et Poirier, 2008b] Belatar, M. et Poirier, F., (2008), Text entry for Mobile devices and users with severe motor impairments : HandiGlyph, a primitive shapes based onscreen keyboard. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers And Accessibility ASSETS'08* (Halifax, Nova Scotia, Canada, October 13 – 15, 2008), ACM, 209–216.
- [Bellengier et al., 2004] Bellengier, E., Blache, P., Rauzy, S., (2004). PCA : un système d'aide à la communication alternatif évolutif et réversible, In *Proceedings of ISAAC'04*, 78-85, 6-8 mai 2004, Neuchâtel, Suisse
- [Bellman et MacKenzie, 1998] Bellman, T., et MacKenzie, I. S. (1998). A probabilistic character layout strategy for mobile text entry. In *Proceedings of Graphics Interface '98*, 168-176. Toronto: Canadian Information Processing Society.
- [Blankertz et al., 2006] Blankertz, B., Dornhege, G., Krauledat, M., Schröder, M., Williamson, J., Murray-Smith, R. et Müller, K.R. (2006). The Berlin Brain-Computer Interface presents the

- novel mental typewriter Hex-o-Spell. In *Proceedings of the 3rd International Brain-Computer Interface Workshop and Training Course'06*. Verlag der Technischen Universität Graz, 2006, 108-109.
- [Blickenstorfer, 1995] Blickenstorfer, C. H. (1995). Graffiti: Wow!!!!, In *Pen Computing Magazine*, Janvier 1995, 30-31.
- [Bloor, 2003] Bloor, R. (2003). Symbian DevZone taks to What Next: A better small device interface?, The Symbian DevZone, June 16, 2003. Disponible à: http://www.wirelessdevnet.com/symbian/rb_57.html, visité le 12/08/2008.
- [Boisière et Dours, 2000] Boisière, Ph. et Dours, D. (2000). VITIPI : Un système d'aide à l'écriture basé sur un principe d'auto-apprentissage et adapté à tous les handicaps moteurs, In *Proceedings of Handicap'00*, IFRATH, Paris, Juin 2000, 81-86
- [Bouteruche et al., 2005] Bouteruche, F., Deconde, G., Anquetil, E., et Jamet, E. (2005). Design and evaluation of handwriting input interfaces for small-size mobile devices, In *Proceedings of the 1st Workshop on Improving and Assessing Pen-Based Input Techniques*, (Edinburgh, Scotland, 2005), 49-56.
- [Brouard, 2004] Brouard, C. (2004). Le handicap en chiffres, CTNERHI, disponible à l'adresse : <http://www.sante.gouv.fr/drees/handicap/handicap.pdf>
- [Cantegrit et Toulotte, 2001] Cantegrit, B., Toulotte J.M., (2001). Réflexions sur l'aide à la communication des personnes présentant un handicap moteur, *atelier Ingénierie de la langue et handicap*, TALN'2001, Tours, France. 2001. pp 193-202.
- [Cao et Zhai, 2007] Cao, X. et Zhai, S. (2007). Modeling human performance of pen stroke gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 1495-1504. DOI= <http://doi.acm.org/10.1145/1240624.1240850>
- [Card et al., 1980] Card, S.K., Moran, T.P., et Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23, 7 (Jul. 1980), 396-410. DOI= <http://doi.acm.org/10.1145/358886.358895>
- [Card et al., 1983] Card, S.K., Newell, A., et Moran, T.P. (1983). *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc.
- [Castellucci, 2007] Castellucci, S.J. (2007). *Mobile Text Entry Using Ambiguous Keypads: New Metrics In A New Toolkit*, Master Thesis, YORK UNIVERSITY. Available at: <http://www.cse.yorku.ca/~steven/c/docs/Thesis.pdf>
- [Castellucci et MacKenzie, 2008] Castellucci, S. J. et MacKenzie, I. S. (2008). Unigest: text entry using three degrees of motion. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems* (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 3549-3554. DOI= <http://doi.acm.org/10.1145/1358628.1358889>
- [Ceccato et al., 2001] Ceccato, F. Delcey, M. et Papa, A. (2001). Personnes déficientes motrices. Personnes handicapées motrices vivant en institution : une population introuvable dans l'enquête HID ?, In *Document de travail : Premiers travaux d'exploitation de l'enquête HID - Colloque scientifique*, Montpellier, 30 novembre et 1er décembre 2000", Série Études n°16, juillet, 2001, 209-229.
- [Chang et al., 1992] Chang, S.K., Costagliola, G., Orefice, S., Polese, G. et Baker, B.R. (1992). A methodology for iconic language design with application to augmentative communication. In *Proceedings of the IEEE Workshop on Visual Languages* in Seattle, Washington, IEEE Computer Society Press, 110-116. DOI=10.1109/WVL.1992.275776
- [Chau et al., 2006] Chau, D.H., Wobbrock, J.O., Myers, B.A., et 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, 2006). CHI '06. ACM, New York, NY, 640-645. DOI= <http://doi.acm.org/10.1145/1125451.1125583>
- [Chen et Gong, 1984] Chen, C.K. et Gong, R.W. (1984). Evaluation of Chinese input methods. In *Proceedings of Computer Processing of Chinese and Oriental Languages* (Nov. 1984). 1(4), 236-247.

-
- [Cockburn et Siresena, 2003] 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.
- [Coutaz, 1998] Coutaz, J. (1998). Interfaces Homme-Machine : "le Futur ne Manque pas d'Avenir" , In *conférence invitée ERGO-IA'98*, Proc. ERGO-IA'98, Ed. ESTIA/ILS.
- [Cox et al., 2007] Cox, A.L., Cairns, P.A., Walton, A. et Lee, S. (2007). Tlk or txt? Using voice input for SMS composition, In *Personal and Ubiquitous Computing*, 12(8), 567-588, DOI=10.1007/s00779-007-0178-8
- [Diaper et Stanton, 2003] Diaper, D et Stanton, N. (2003). *The handbook of task analysis for human-computer interaction*, Lawrence Erlbaum Associates, ISBN 0805844325, 9780805844320
- [Dunlop et Crossan, 2000] Dunlop, M.D., Crossan, A., (2000). Predictive text entry methods for mobile phones, In *Personal and Ubiquitous Computing*, 4(2), 134-143, DOI=<http://dx.doi.org/10.1007/BF01324120>
- [Dunlop, 2004] Dunlop, M.D. (2004). Watch-top text-entry: can phone-style predictive text-entry work with only 5 buttons?, In *Proceedings of HCI 2004*, ACM Press.
- [Evreinov et Raisamo, 2003] Evreinov, G., Raisamo, R. (2003). An exploration of vibration during interaction with polarized texture. In *Proceedings of the Third International Conference on Modeling & Experimental Measurements in Acoustics. Acoustics III*, D. June, Cadiz, Spain, WIT Press UK, 2003, 309-316.
- [Fitts, 1954] Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, volume 47, number 6, June 1954, pp. 381-391. (Reprinted in *Journal of Experimental Psychology: General*, 121(3):262-269, 1992).
- [Fleetwood et al., 2002] Fleetwood, M.D. Byrne, M.D. Centgraf, P. Dudziak, K.Q. Lin, B. et Mogilev, D. (2002). An evaluation of text entry in Palm OS: Graffiti and virtual keyboard, in *Proceedings of HFES*, 2002, 617-621.
- [Furnas, 1986] Furnas, G.W. (1986). Generalized fisheye views. In *Proceedings of SIGCHI Bull.* 17, 4 (Apr. 1986), 16-23.
- [Ghedira et al., 2006] Ghedira, S., Pino, P., Bourhis G. (2006). Interaction personne handicapée/système à balayage : vers un réglage automatique de la vitesse de défilement adapté à l'utilisateur. In *Proceedings of ERGO-IA'06*, 317-320, Biarritz, 2006.
- [Goldberg et Richardson, 1993] Goldberg, D. et Richardson, C. (1993). Touch-typing with a stylus. In *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, 80-87. New York: ACM.
- [Goldstein et al., 1999] Goldstein, M., Book, R., Alsiö, G., et Tessa, S. (1999). Non-keyboard QWERTY touch typing: a portable input interface for the mobile user. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI Is the Limit* (Pittsburgh, Pennsylvania, United States, May 15 - 20, 1999). CHI '99. ACM, New York, NY, 32-39. DOI= <http://doi.acm.org/10.1145/302979.302984>
- [Gong et al., 2005] Gong, J., Haggerty, B., et 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, 2005). CHI '05. ACM, New York, NY, 1399-1402. DOI= <http://doi.acm.org/10.1145/1056808.1056926>
- [Gong et Tarasewich, 2005] Gong, J. et Tarasewich, P. (2005). Testing Predictive Text Entry Methods with Constrained Keypad Designs. In *Proceedings of HCI'05 International Conference*, disponible à : http://www.ccs.neu.edu/home/tarase/HCI05_GongTarase.pdf
- [Gong et Tarasewich, 2006] Gong, J. et Tarasewich, P. (2006). A new error metric for text entry method evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI '06. ACM, New York, NY, 471-474. DOI= <http://doi.acm.org/10.1145/1124772.1124843>

- [Goodenough-Trepagnier et al., 1986] Goodenough-Trepagnier, C., Rosen, M.J. et Galdieri, B. (1986). Word menu reduces communication rate. In *Proceedings of the 9th Annual Conference of the Rehabilitation Engineering and Assistive Technology Society of North America (RESNA '86)*. Minneapolis, Minnesota (June 23-26, 1986). Arlington, Virginia: RESNA Press, 354-356.
- [Grenier, 2000] Grenier, J.G., (2000). *Le dictionnaire d'informatique et d'Internet (Anglais-Français)*, Édition de la maison du dictionnaire.
- [Grover et al., 1998] Grover, D.L., King, M.T. et Kushler, C.A., (1998). Reduced keyboard disambiguating computer. *U.S. Patent no. 5,818,437*.
- [Harbusch et al., 2003] Harbusch, K. Hasan, S. Hoffmann, H. Kühn, M. et Schüler B. (2003). Domain-Specific Disambiguation for Typing with Ambiguous Keyboards. In *Proceedings of the Workshop on Language Modeling for Text Entry Methods at the 10th Conference of the European Chapter of the Association for Computational Linguistics EACL'03*, (Budapest, Hungary, April 12 – 17, 2003). 67-74.
- [Harris et Vanderheiden, 1980] Harris, D., et Vanderheiden, G.C. (1980). *Augmentative Communication Techniques*, R.L. Schiefelbusch ed. Baltimore: University Park Press.
- [Hasselgren et al., 2003] Hasselgren, J., Montnemery, E., et Svensson, M. (2003). HMS: A predictive text entry method using bigrams. In *Extended Abstracts 10th Conference of the European Chapter of the Association of Comp. Linguistics* (2003), 43-49.
- [He et al., 2006] He, G., Pan, Z., Quarre, C., Zhang, M., Xu, H. (2006). Multi-stroke Freehand Text Entry Method Using OpenVG and Its Application on Mobile Devices, In *Technologies for E-Learning and Digital Entertainment*, Springer, 791-796, DOI=10.1007/11736639_97, available at : <http://www.springerlink.com/content/r3l124w3221640h0>
- [Hick, 1952] Hick, W.E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, No 4, 11-26.
- [Hornbaek et Frokjaer, 2001] Hornbaek, K., et Frokjaer, E., (2001). Reading of Electronic Documents: The Usability of Linear, Fisheye, and Overview+Detail Interfaces, In *Proceedings of the ACM CHI'01*, 293-300.
- [Hsu, 1991] Hsu, S.C. (1991). A flexible Chinese character input scheme. In *Proceedings of the 4th Annual ACM Symposium on User interface Software and Technology* (Hilton Head, South Carolina, United States, November 11 - 13, 1991). UIST '91. ACM, New York, NY, 195-200. DOI=<http://doi.acm.org/10.1145/120782.120803>
- [Hyman, 1953] Hyman, R. (1953). Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*, 45(3), 188-96.
- [Isokoski et Raisamo, 2000] Isokoski, P. et Raisamo, R. (2000). Device independent text input: A rationale and an example. In *Proceedings of AVI '00*. ACM Press, 76-83.
- [Isokoski, 1999] Isokoski, P. (1999). *A minimal device-independent text input method*. Unpublished MSc Thesis, University of Tampere, Tampere Finland. Disponible sur : <http://www.cs.uta.fi/~poika/g/>, visité le 07/12/2008.
- [Isokoski, 2004] Isokoski, P. (2004). *Manual text entry: Experiments, Models, and Systems*. Report A-2004-3, Department of Computer Sciences, FIN-33014, University of Tampere, Finland. Available at: http://www.cs.uta.fi/~poika/vk/isokoski_thesis_complete.pdf
- [James et Reischel, 2001] James, C. L. et 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* (Seattle, Washington, United States). CHI '01. ACM, New York, NY, 365-371. DOI= <http://doi.acm.org/10.1145/365024.365300>
- [Jhaveri, 2003] Jhaveri, N. (2003). Two Characters per Stroke - A Novel Pen-Based Text Input Technique, in Grigori Evreinov (ed.), *New Interaction Techniques 2003* (Report B-2003-5), Department of Computer and Information Sciences, University of Tampere (2003) 10-15, Available at: <http://www.cs.uta.fi/reports/bsarja/B-2003-5.pdf>
- [Kano et al., 2007] Kano, A., Read, J. C., Dix, A., et MacKenzie, I. S. (2007). ExpECT: An expanded error categorization method for text input. In *Proceedings of People and Computers XXI: Proceedings of HCI 2007*, 149-156. London: British Computer Society.

-
- [Karlson et Bederson, 2007] Karlson, A.K. et Bederson, B.B. (2007). ThumbSpace: generalized one-handed input for touchscreen-based mobile devices. In *Proceedings of INTERACT 2007*, Springer (2007), 324-338.
- [Kober et al., 2001] Kober, H. Skepner, E. Jones, T. Gutowitz, H. et MacKenzie, I.S. (2001). Linguistically optimized text entry on a Mobile Phone, disponible sur www.eatoni.com/research/chi.pdf
- [Költringer et Grechenig, 2004] Költringer, T., et 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, 2004). CHI '04. ACM, New York, NY, 1175-1178. DOI= <http://doi.acm.org/10.1145/985921.986017>
- [Kristensson et Zhai, 2004] Kristensson, P.O., et Zhai, S. (2004). SHARK2: a large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (UIST '04). ACM Press: 43-52.
- [Kristensson et Zhai, 2007] Kristensson, P.O., et Zhai, S. (2007). Learning shape writing by game playing. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems* (San Jose, CA, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 1971-1976. DOI= <http://doi.acm.org/10.1145/1240866.1240934>
- [Lagoá et al., 2007] Lagoá, P. Santana, P. Guerreiro, T. Goncalves, D. Jorge, J. (2007). BloNo: A New Mobile Text-Entry Interface for the Visually Impaired, *Lecture Notes In Computer Science*, no 4555, pages 908-917. DOI= http://dx.doi.org/10.1007/978-3-540-73281-5_99
- [Lamping et al., 1995] Lamping, J., Rao, R., et Pirolli, P. (1995). A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Denver, Colorado, United States, May 07 - 11, 1995). I. R. Katz, R. Mack, L. Marks, M. B. Rosson, and J. Nielsen, Eds. Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co., New York, NY, 401-408. DOI= <http://doi.acm.org/10.1145/223904.223956>
- [Lesh et al., 2000] Lesh, G.W., Higginbotham, D.J., et Moulton, B.J. (2000). Techniques for automatically updating scanning delays. In *Proceedings of Rehabilitation Engineering and Assistive Technology Society of North America RESNA'00* (Orlando, FL, USA, 28 June - 2 July 2000), 85-87.
- [Levenshtein, 1966] Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10, 707-710.
- [Levy, 2002] Levy, D. (2002). The Fastap Keypad and Pervasive Computing. In *Proceedings of the First international Conference on Pervasive Computing* (August 26 - 28, 2002). F. Mattern and M. Naghshineh, Eds. Lecture Notes In Computer Science, vol. 2414. Springer-Verlag, London, 58-68.
- [Likert, 1932] Likert, R., (1932). A technique for the measurement of attitude, *Archives of Psychology*, no. 140, 44-45.
- [Luo et John, 2005] Luo, L. et John, B.E. (2005). Predicting task execution time on handheld devices using the keystroke-level model. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems* (Portland, OR, USA, April 02 - 07, 2005). CHI '05. ACM, New York, NY, 1605-1608. DOI= <http://doi.acm.org/10.1145/1056808.1056977>
- [Luoma, 2003] Luoma, M. (2003). Symbol Creator: Usability Evaluation of the Novel Pen-Based Text Input Technique. *Technical Notes in New Interaction Techniques*, University of Tampere. Disponible à : http://www.cs.uta.fi/~grse/NIT_2003/Student_prjs/Melind_L/SymbolCreator.pdf
- [Lyons et al., 2004] Lyons, K., Starner, T., Plaisted, D., Fusia, J., Lyons, A., Drew, A., et Looney, E. W. (2004). Twiddler typing: one-handed chording text entry for mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, 671-678. DOI= <http://doi.acm.org/10.1145/985692.985777>
- [MacKenzie et al., 1991] MacKenzie, I.S., Sellen, A., et Buxton, W.A. (1991). A comparison of input devices in element pointing and dragging tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology* (New Orleans,

- Louisiana, United States, April 27 - May 02, 1991). S. P. Robertson, G. M. Olson, and J. S. Olson, Eds. CHI '91. ACM, New York, NY, 161-166. DOI= <http://doi.acm.org/10.1145/108844.108868>
- [MacKenzie et al., 1999] MacKenzie, I.S., Zhang, S.X., et Soukoreff, R.W. (1999). Text entry using soft key-boards. *Behaviour & Information Technology*, 18, 235-244.
- [MacKenzie et al., 2001] MacKenzie, I.S., Kober, H., Smith, D., Jones, T., et Skepner, E. (2001). LetterWise: prefix-based disambiguation for mobile text input. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (Orlando, Florida, November 11 - 14, 2001). UIST '01. ACM, New York, NY, 111-120. DOI= <http://doi.acm.org/10.1145/502348.502365>
- [MacKenzie et al., 2006] MacKenzie, I.S., Chen, J., et Oniszcak, A. (2006). Unipad: single stroke text entry with language-based acceleration. In *Proceedings of the 4th Nordic Conference on Human-Computer interaction: Changing Roles* (Oslo, Norway, October 14 - 18, 2006). A. Mørch, K. Morgan, T. Bratteteig, G. Ghosh, and D. Svanaes, Eds. NordiCHI '06, vol. 189. ACM, New York, NY, 78-85. DOI= <http://doi.acm.org/10.1145/1182475.1182484>
- [MacKenzie et Buxton, 1992] MacKenzie, I.S. et Buxton, W. (1992). Extending Fitts' law to two-dimensional tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Monterey, California, United States, May 03 - 07, 1992). P. Bauersfeld, J. Bennett, and G. Lynch, Eds. CHI '92. ACM, New York, NY, 219-226. DOI= <http://doi.acm.org/10.1145/142750.142794>
- [MacKenzie et Soukoreff, 2002] MacKenzie, I.S. et Soukoreff, R.W. (2002). Text Entry for Mobile Computing: Models and Methods, Theory and Practice. *Human-Computer Interaction*, 17 (2), 147-198. Available at: http://www.informaworld.com/10.1207/S15327051HCI172&3_2
- [MacKenzie et Soukoreff, 2003] MacKenzie, I.S. et Soukoreff, R.W. (2003). Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, April 05 - 10, 2003). CHI '03. ACM, New York, NY, 754-755. DOI= <http://doi.acm.org/10.1145/765891.765971>
- [MacKenzie et Tanaka-Ishii, 2007] MacKenzie, I. S., et Tanaka-Ishii, K. (2007). *Text entry systems: Mobility, accessibility, universality*, Morgan Kaufmann, San Francisco-CA, ISBN=978-0123735911
- [MacKenzie et Zhang, 1999] MacKenzie, I.S. et Zhang, S.X. (1999). The design and evaluation of a high-performance soft keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI Is the Limit* (Pittsburgh, Pennsylvania, United States, May 15 - 20, 1999). CHI '99. ACM, New York, NY, 25-31. DOI= <http://doi.acm.org/10.1145/302979.302983>
- [MacKenzie, 1991] MacKenzie, I.S. (1991). *Fitts' law as a performance model in human-computer interaction*. Doctoral dissertation. University of toronto: toronto, ontario, canada.
- [MacKenzie, 1992] MacKenzie, I.S. (1992). Fitts' law as a research and design tool. In *Proceedings of Human-Computer Interaction, HCI'92*, vol. 7, 91-139
- [MacKenzie, 1995] MacKenzie, I.S. (1995). Input devices and interaction techniques for advanced computing. In *Virtual Environments and Advanced interface Design*, W. Barfield and T. A. Furness, Eds. Oxford University Press, New York, NY, 437-470.
- [MacKenzie, 2002] MacKenzie, I.S. (2002). Mobile text entry using three keys. In *Proceedings of the Second Nordic Conference on Human-Computer interaction* (Aarhus, Denmark, October 19 - 23, 2002). NordiCHI '02, vol. 31. ACM, New York, NY, 27-34. DOI= <http://doi.acm.org/10.1145/572020.572025>
- [Magnien et al., 2003] Magnien, L., Bouraoui, J.L. et Vella, F. (2003). Utilisation d'indices visuels pour l'aide à la saisie de texte sur PDA. In *Proceedings of 15èmes Conférence Francophone sur l'Interaction Homme-Machine (IHM 2003)*, Caen, 24-NOV-03-28-NOV-03, ACM, IBSN: 1-58113-803-2/03/0011, New York, 252-255, novembre 2003.
- [Mankoff et Abowd, 1998] Mankoff, J., et Abowd, G. A. (1998). Cirrin: A word-level Unistroke keyboard for pen input, In *Proceedings of the ACM Symposium on User Interface Software and Technology - UIST '98*, 213-214, New York: ACM.

-
- [Martin et Pecci, 2007] Martin, B., Pecci, I., (2007). État de l'art des claviers physiques et logiciels pour la saisie de texte, *Revue d'Interaction Homme-Machine*, 8(2), 147-205.
- [Martin, 2005] Martin, B. (2005). VirHKey: un clavier gestuel hyperbolique avec retour visuel. In *Proceedings of the 17th international Conference on Francophone Sur L'interaction Homme-Machine* (Toulouse, France, September 27 - 30, 2005). IHM 2005, vol. 264. ACM, New York, NY, 19-26. DOI= <http://doi.acm.org/10.1145/1148550.1148553>
- [Masui, 1998] Masui, T. (1998). An efficient text input method for pen-based computers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Los Angeles, California, United States, April 18 - 23, 1998). C. Karat, A. Lund, J. Coutaz, and J. Karat, Eds. Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co., New York, NY, 328-335. DOI= <http://doi.acm.org/10.1145/274644.274690>
- [Matias et al., 1993] Matias, E., MacKenzie, I. S., et Buxton, W. (1993). Half-QWERTY: a one-handed keyboard facilitating skill transfer from QWERTY. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (Amsterdam, The Netherlands, April 24 - 29, 1993). CHI'93. ACM, New York, NY, 88-94. DOI= <http://doi.acm.org/10.1145/169059.169097>
- [Miniotos et al., 2003] Miniotos, D., Špakov, O., et Evreinov, G. (2003). Symbol Creator: An Alternative Eye-Based Text Entry Technique With Low Demand For Screen Space. In *Proceedings of INTERACT'03* (Zurich, Switzerland, September 1 – 5, 2003), IOS Press, IFIP, 137-143.
- [Myers et Wobbrock, 2005] Myers, B., et Wobbrock, J.O. (2005). Text Input to Handheld Devices for People with Physical Disabilities. In *Proceedings of Human-Computer Interaction International Conference HCII'05* (Las Vegas, Nevada, USA, July 22 – 27, 2005), NJ: Lawrence Erlbaum Associates Publisher, publication électronique.
- [Nesbat, 2003] Nesbat, S. B. (2003). A system for fast, full-text entry for small electronic devices. In *Proceedings of the 5th international Conference on Multimodal interfaces* (Vancouver, British Columbia, Canada, November 05 - 07, 2003). ICMI '03. ACM, New York, NY, 4-11. DOI= <http://doi.acm.org/10.1145/958432.958437>
- [New et al., 2007] New, B., Brysbaert, M., Veronis, J., et Pallier, C. (2007). The use of film subtitles to estimate word frequencies. *Applied Psycholinguistics*, 28(4), 661-677.
- [Pavlovych et Stuerzlinger, 2005] Pavlovych, W. et Stuerzlinger (2005). An Analysis of Novice Text Entry Performance on Large Interactive Wall Surfaces, In *Proceedings of HCI International HCII'05*, Ed. G. Salvendy, Lawrence Erlbaum, ISBN 0-8058-5807-5, CD-ROM.
- [Perlin, 1998] Perlin, K. (1998). Quikwriting: Continuous stylus-based text entry. In *Proceedings of the ACM Symposium on User Interface Software and Technology - UIST '98* (San Fransisco, CA, USA, November 1 – 4, 1998), ACM, 215-216.
- [Perraud et al., 2003] Perraud, C. Viard-Gaudin, E.M., et Lallican, P.M. (2003). N-gram and n-class models for on line handwriting recognition. In *Proceedings of the 7th International Conference on Document Analysis and Recognition ICDAR'03*, (Edinburgh, Scotland, August 3 – 6, 2003), vol. 2, 1053–1057.
- [Poirier et Belatar, 2006a] Poirier, F. et Belatar, M. (2006). Glyph 2: une saisie de texte avec deux appuis de touche par caractère - principes et comparaisons. In *Proceedings of the 18th international Conference of the Association Francophone D'interaction Homme-Machine IHM'06* (Montreal, Canada, April 18 - 21, 2006), vol. 133. ACM, New York, NY, 159-162. DOI= <http://doi.acm.org/10.1145/1132736.1132758>
- [Poirier et Belatar, 2006b] Poirier F. et Belatar M., (2006), Evaluation d'analogies scripturales pour la conception d'une méthode de saisie en mobilité, Uni-Glyph, In *Proceedings of Ergo'IA 06*, (Biarritz, France, October 15 – 17, 2006), ESTIA & ESTIA.INNOVATION (ed.), p. 333-336.
- [Poirier et Belatar, 2007] Poirier F. et Belatar M., (2007), UniGlyph: only one keystroke per character on a 4-button minimal keypad for key-based text entry, In *HCI International 07 Posters*, (Beijing, P.R. China, July 22 – 27, 2007), Springer, p. 479-483.

- [Poirier et Sad, 2007] Poirier, F. et Sad, H. (2007). A New Performance Measure Taking into Account the Mental Load in Mobile Text Entry Tasks, In *Proceedings of INTERACT'07*, C. Baranauskas et al. (Eds.), LNCS 4663, Part II, 653 – 656, 2007.
- [Poirier et Sad, 2008] Poirier, F. et Sad, H. (2008). A Platform for Mobile Text Entry Methods Evaluation, In *Proceedings of the International Conference on Advances in Computer-Human Interaction ACHI'08* (Sainte Luce, Martinique, February 10 – 15, 2008), 183-188.
- [Proschowsky et al., 2006] Proschowsky, M., Schultz, N., et Jacobsen, N. E. (2006). An intuitive text input method for touch wheels. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI '06. ACM, New York, NY, 467-470. DOI= <http://doi.acm.org/10.1145/1124772.1124842>
- [Raynal, 2006] Raynal, M. (2006). Claviers GAG : claviers logiciels optimisés pour la saisie de texte au stylet. In *Proceedings of 18th international conference of Association Francophone d'Interaction Homme-Machine (IHM)*, Montréal, Canada, 18/04-21/04, vol. 133, ACM Press, 3-10.
- [Raynal, 2007] Raynal, M. (2007). Le système KeyGlass : Système d'ajout dynamique de touches sur clavier logiciel. *Traitement Automatique des Langues*, Association pour le Traitement Automatique des Langues (ATALA), Numéro spécial Communication assistée, 48(2), 97-121.
- [Raynal et Truillet, 2006] Raynal, M. et Truillet, P. (2006). Le clavier FishEye. Dans : *Interaction Homme-Machine (IHM 2006)*, Montréal, Canada, 18/04/06-21/04/06, Vol. 133, ACM Press, p. 289-290, 2006.
- [Roeber, 2003] Roeber, H., Bacus, J. et Tomasi, C. (2003). Typing in thin air: the canesta projection keyboard - a new method of interaction with electronic devices. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, April 05 - 10, 2003). CHI '03. ACM, New York, NY, 712-713. DOI= <http://doi.acm.org/10.1145/765891.765944>
- [Rosenberg et Tunney, 2008] Rosenberg, J. et Tunney, R.J. (2008). Human vocabulary use as display, *Evolutionary Psychology*, 2008, 6(3), 538-549
- [Sad et Poirier, 2008] Sad, H. et Poirier, F. (2008). Evaluation and optimization of word disambiguation for text entry methods. In *Proceedings of the 6th Computer-Aided Design of User Interfaces CADUI'08* (Albacete, Espagne, June 11 – 13, 2008), 111-120.
- [Sadun, 2008] Sadun, E. (2008). Interacting with Your iPod Touch , *Taking Your iPod touch to the Max*, Apress Eds., 19-36, DOI=10.1007/978-1-4302-0642-2_2, Disponible à www.springerlink.com/content/q3371674752p8140
- [Sagawa et al., 1997] Sagawa, H., Takeuchi, M., et Ohki, M. (1997). Description and recognition methods for sign language based on gesture components. In *Proceedings of the 2nd international Conference on intelligent User interfaces* (Orlando, Florida, United States, January 06 - 09, 1997). J. Moore, E. Edmonds, and A. Puerta, Eds. IUI '97. ACM, New York, NY, 97-104. DOI= <http://doi.acm.org/10.1145/238218.238310>
- [Sarkar et Brown, 1993] Sarkar, M. et Brown, M. H. (1993). *Graphical Fisheye Views*. Technical Report. UMI Order Number: CS-93-40., Brown University.
- [Sazawal et al., 2002] Sazawal, V., Want, R., et Borriello, G. (2002). The Unigesture Approach: One-Handed Text Entry for Small Devices, In *Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction* (Pisa, Italy, September 18-20, 2002), 256-270,
- [Schadle et al., 2004] Schadle, I., Antoine, J-Y., Le Pévédic, B., et Poirier, F. (2004). SibyMot - Modélisation stochastique du langage intégrant la notion de chunks. In *Proceedings of TALN'04*, (Fès, Morocco, April 19 – 22, 2004).
- [Schaffer et al., 1996] Schaffer, D., Zuo, Z., Greenberg, S., Bartram, L., Dill, J., Dubs, S., et Roseman, M., (1996). Navigating Clustered Networks through Fisheye and Full-Zoom Methods, In *ACM Transactions on Computer-Human Interaction*, 3(2), 162-188.
- [Sears et Arora, 2002] Sears, A. et Arora, R. (2002). Data entry for mobile devices: An empirical comparison of novice performance with Jot and Graffiti. *Interacting with Computers*, 14(5), 413-433.

-
- [Shannon, 1948] Shannon, C.E., (1948). A mathematical theory of communication, *The bell system technical journal*, vol. 27, 379–423, 623–656 (two parts, July and October, 1948).
- [Shein et al., 1991] Shein, F., Hamann, G., Brownlow, N., Treviranus, J., Milner, M. et Parnes, P. (1991). WiViK: A visual keyboard for Windows 3.0. In *Proceedings of the 14th Annual Conference of the Rehabilitation Engineering and Assistive Technology Society of North America* (RESNA '91). Kansas City, Missouri, June 1991, 160-162.
- [Shu, 2003] Shu, C. F. (2003). New Products. *IEEE MultiMedia*, 10(4), p. 3, ISSN: 1070-986X
- [Smith et al., 1996] Smith, A., Dunaway, J., Demasco, P., et Peischl, D. (1996). Multimodal input for computer access and augmentative communication. In *Proceedings of the Second Annual ACM Conference on Assistive Technologies* (Vancouver, British Columbia, Canada, April 11 - 12, 1996). Assets '96. ACM, New York, NY, 80-85. DOI= <http://doi.acm.org/10.1145/228347.228361>
- [Smith, 2000] Smith, J.M.D. (2000). Thumbscript™ : Designing a general solution to the problem of text input in small devices. White Paper, Thumbscript Development 2000, Disponible à : <http://www.thumbscript.com/Downloads/TSWhitePaper.pdf>, visitée le 06/08/2008.
- [Soukoreff et MacKenzie, 1995] Soukoreff, R. W. et MacKenzie, I. S. (1995). Generalized Fitts' law model builder. In *Conference Companion on Human Factors in Computing Systems* (Denver, Colorado, United States, May 07 - 11, 1995). I. Katz, R. Mack, and L. Marks, Eds. CHI '95. ACM, New York, NY, 113-114. DOI= <http://doi.acm.org/10.1145/223355.223456>
- [Soukoreff et MacKenzie, 2001] Soukoreff, R.W. et MacKenzie, I.S. (2001). Measuring errors in text entry tasks: an application of the Levenshtein string distance statistic. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems* (Seattle, Washington, March 31 - April 05, 2001). CHI '01. ACM, New York, NY, 319-320. DOI= <http://doi.acm.org/10.1145/634067.634256>
- [Soukoreff et MacKenzie, 2003] Soukoreff, R.W. et MacKenzie, I.S. (2003). Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, April 05 - 10, 2003). CHI '03. ACM, New York, NY, 113-120. DOI= <http://doi.acm.org/10.1145/642611.642632>
- [Soukoreff et MacKenzie, 2004] Soukoreff, R.W. et MacKenzie, I.S. (2004). Recent developments in text-entry error rate measurement. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, 1425-1428. DOI= <http://doi.acm.org/10.1145/985921.986081>
- [Stolcke, 2002] Stolcke, A. (2002). SRILM -- an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing* (Denver, CO, USA, September 16 – 12, 2002), 901-904, <http://www.speech.sri.com/projects/srilm/>
- [Tappert et al., 1990] Tappert, C.C., Suen, C.Y., et Wakahara, T. (1990). The state of the art in on-line handwriting recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, 787-808.
- [Tarasewich, 2003] Tarasewich, P. (2003). Evaluation of thumbwheel text entry methods. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, April 05-10, 2003). CHI'03. ACM, New York, NY, 756-757. DOI= <http://doi.acm.org/10.1145/765891.765972>
- [Tinwala et MacKenzie, 2008] Tinwala, H. et MacKenzie, I. S. (2008). Letterscroll: text entry using a wheel for visually impaired users. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems* (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 3153-3158. DOI= <http://doi.acm.org/10.1145/1358628.1358823>
- [Venolia et Neiberg, 1994] Venolia, D., et Neiberg, F. (1994). T-Cube: a fast, self-disclosing pen-based alphabet. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Celebrating interdependence* (Boston, Massachusetts, United States, April 24 - 28, 1994). B. Adelson, S. Dumais, and J. Olson, Eds. CHI'94. ACM, New York, NY, 265-270. DOI= <http://doi.acm.org/10.1145/191666.191761>

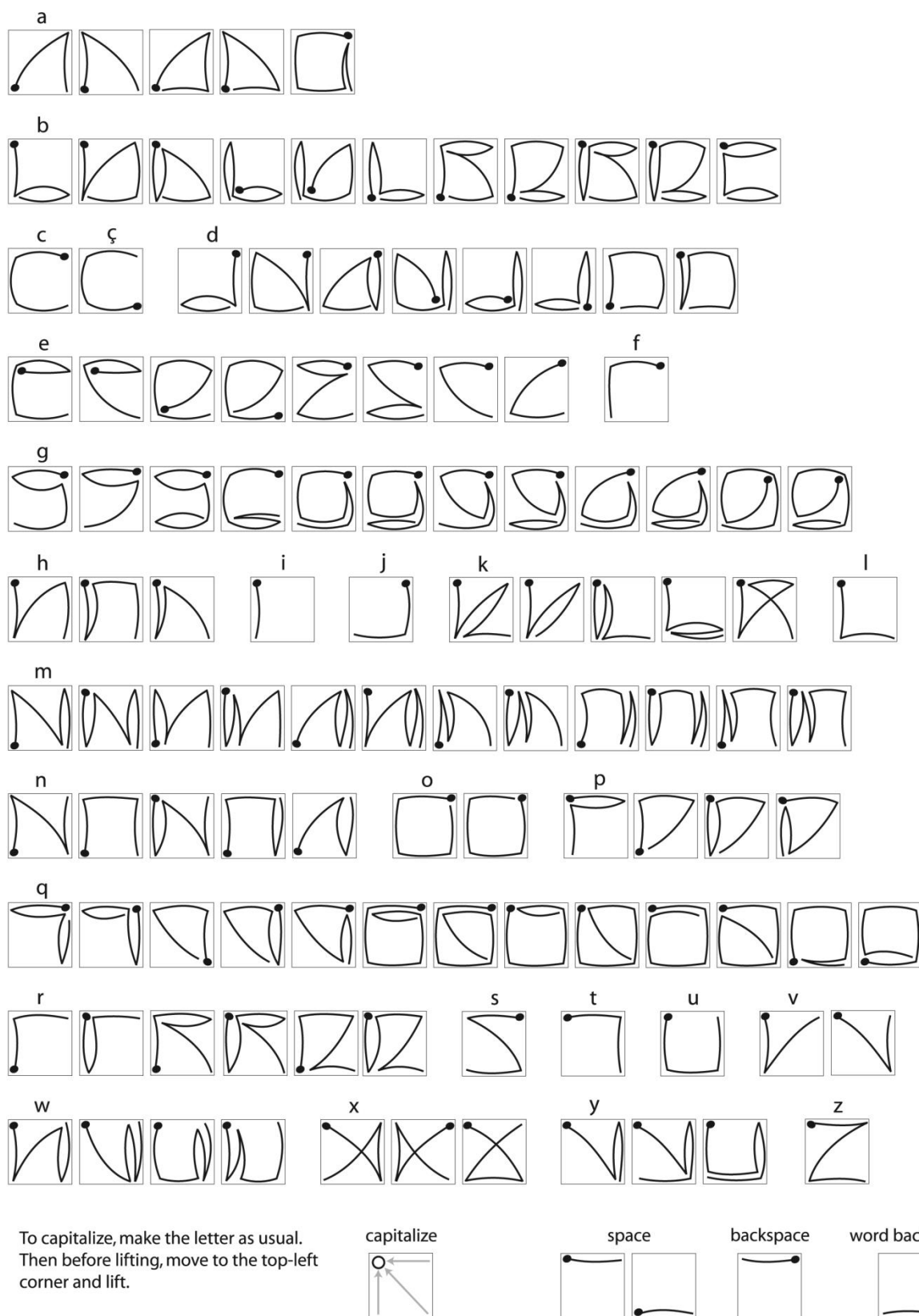
- [Wandmacher et al., 2007] Wandmacher, T., Béchet, N., Barhoumi, Z., Poirier, F. et Antoine, J-Y. (2007). Système Sibylle d'aide à la communication pour personnes handicapées : modèle linguistique et interface utilisateur. In *Atelier : Reconstruire la langue dans les communications alternatives et augmentées, TALN'07* (Toulouse, France, June 5 – 8, 2007), vol. 2, 539-548.
- [Wandmacher et al., 2008] Wandmacher, T., Antoine, J., Poirier, F., et Départe, J. (2008). Sibylle, An Assistive Communication System Adapting to the Context and Its User. In *ACM Transactions on Accessible Computing* 1(1) (May. 2008), 1-30. DOI= <http://doi.acm.org/10.1145/1361203.1361209>
- [Ward et al., 2000] Ward, D.J., Blackwell, A.F., et MacKay, D.J. (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* (San Diego, California, United States, November 06 - 08, 2000). UIST '00. ACM, New York, NY, 129-137. DOI= <http://doi.acm.org/10.1145/354401.354427>
- [Wigdor et Balakrishnan, 2003] Wigdor, D. et 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* (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, 81-90. DOI= <http://doi.acm.org/10.1145/964696.964705>
- [Wigdor et Balakrishnan, 2004] Wigdor, D. et 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* (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, 81-88. DOI= <http://doi.acm.org/10.1145/985692.985703>
- [Williamson et Murray-Smith, 2005] Williamson, J., et Murray-Smith, R. (2005). Dynamics and probabilistic text entry, *Hamilton Summer School on Switching and Learning in Feedback systems* (Murray-Smith R and Shorten R, eds.), Springer-Verlag, LNCS, Vol. 3355, p333-342.
- [Wobbrock et al., 2003] Wobbrock, J.O., Myers, B.A. et Kembel, J.A. (2003). EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the ACM Symposium on User Interface Software and Technology UIST '03.* (Vancouver, British Columbia, November 2-5, 2003). New York: ACM Press, 61-70.
- [Wobbrock et al., 2004a] Wobbrock, J.O., Myers, B.A., et Aung, H.H. (2004). Writing with a joystick: a comparison of date stamp, selection keyboard, and EdgeWrite. In *Proceedings of Graphics Interface GI'04* (London, Ontario, Canada, May 17 - 19, 2004). ACM International Conference Proceeding Series, vol. 62. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, 1-8.
- [Wobbrock et al., 2004b] Wobbrock, J.O., Myers, B.A., Aung, H.H., et LoPresti, E.F. (2004). Text entry from power wheelchairs: edgewrite for joysticks and touchpads. In *Proceedings of the 6th international ACM SIGACCESS Conference on Computers and Accessibility* (Atlanta, GA, USA, October 18 - 20, 2004). Assets'04. ACM, New York, NY, 110-117. DOI= <http://doi.acm.org/10.1145/1028630.1028650>
- [Wobbrock et al., 2005] Wobbrock, J.O., Aung, H.H., Myers, B.A. et Lopresti, E.F. (2005). Integrated text entry from power wheelchairs. *Behaviour & Information Technology*, 24(3), 187-203. Retrieved November 13, 2008, disponible à <http://www.informaworld.com/10.1080/01449290512331321729>
- [Wobbrock et al., 2006] Wobbrock, J.O., Myers, B.A., et Rothrock, B. (2006). Few-key text entry revisited: mnemonic gestures on four keys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI '06. ACM, New York, NY, 489-492. DOI= <http://doi.acm.org/10.1145/1124772.1124846>
- [Wobbrock et al., 2007] Wobbrock, J.O., Chau, D.H., et Myers, B.A. (2007). An alternative to push, press, and tap-tap-tap: gesturing on an isometric joystick for mobile phone text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA, April 28 - May 03, 2007). CHI'07. ACM, New York, NY, 667-676. DOI= <http://doi.acm.org/10.1145/1240624.1240728>
- [Wobbrock et al., 2008] Wobbrock, J.O., Cutrell, E., Harada, S., et MacKenzie, I. (2008). An error model for pointing based on Fitts' law. In *Proceeding of the Twenty-Sixth Annual SIGCHI*

-
- Conference on Human Factors in Computing Systems* (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 1613-1622. DOI= <http://doi.acm.org/10.1145/1357054.1357306>
- [Wobbrock et Myers, 2006a] Wobbrock, J.O. et Myers, B. A. (2006). Analyzing the input stream for character- level errors in unconstrained text entry evaluations. In *ACM Transactions on Computer-Human Interaction* (Dec. 2006), 13(4), 458-489. DOI= <http://doi.acm.org/10.1145/1188816.1188819>
- [Wobbrock et Myers, 2006b] Wobbrock, J.O. et Myers, B.A. (2006). Trackball text entry for people with motor impairments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI '06. ACM, New York, NY, 479-488. DOI= <http://doi.acm.org/10.1145/1124772.1124845>
- [Wobbrock, 2003] Wobbrock, J.O. (2003). The benefits of physical edges in gesture-making: empirical support for an edge-based unistroke alphabet. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, April 05 - 10, 2003). CHI '03. ACM, New York, NY, 942-943. DOI= <http://doi.acm.org/10.1145/765891.766083>
- [Wobbrock, 2006] Wobbrock, J.O. (2006) *EdgeWrite: A versatile design for text entry and control*. Ph.D Thesis CMU-HCII-06-104, Carnegie Mellon University Dissertation, July 2006. Available at: <http://reports-archive.adm.cs.cmu.edu/anon/hcii/CMU-HCII-06-104.pdf>
- [Wong et al., 2007] Wong, C.Y., Khong, C.W., et Thwaites, H. (2007). Aesthetics and virtual keyboard designs for PDAs. In *Proceedings of the 4th international Conference on Mobile Technology, Applications, and Systems and the 1st international Symposium on Computer Human interaction in Mobile Technology* (Singapore, September 10 - 12, 2007). Mobility '07. ACM, New York, NY, 181-186. DOI= <http://doi.acm.org/10.1145/1378063.1378093>
- [Yatani et Truong, 2007] Yatani, K. et Truong, K.N. (2007). An evaluation of stylus-based text entry methods on handheld devices in stationary and mobile settings. In *Proceedings of the 9th international Conference on Human Computer interaction with Mobile Devices and Services* (Singapore, September 09 - 12, 2007). MobileHCI '07, vol. 309. ACM, New York, NY, 487-494. DOI= <http://doi.acm.org/10.1145/1377999.1378059>
- [Zhai et al., 2000] Zhai, S., Hunter, M., et Smith, B.A. (2000). The metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design. In *Proceedings of the 13th Annual ACM Symposium on User interface Software and Technology* (San Diego, California, United States, November 06 - 08, 2000). UIST'00. ACM, New York, NY, 119-128. DOI= <http://doi.acm.org/10.1145/354401.354424>
- [Zhai et al., 2002a] Zhai, S., Hunter, M., et Smith, B.A., (2002). Performance optimization of virtual keyboards. In *Proceedings of Human-Computer Interaction'02*, 17(2&3), 229-269, Disponible à http://www.informaworld.com/10.1207/S15327051HCI172&3_4
- [Zhai et al., 2002b] Zhai, S., Sue, A., et Accot, J. (2002). Movement model, hits distribution and learning in virtual keyboarding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves* (Minneapolis, Minnesota, USA, April 20 - 25, 2002). CHI '02. ACM, New York, NY, 17-24. DOI= <http://doi.acm.org/10.1145/503376.503381>
- [Zhai et al., 2005] Zhai, S., Kristensson, P., et Smith, B.A. (2005). In search of effective text input interfaces for off the desktop computing, *Interacting with Computers*. 17(3), 229-250. Disponible à : <http://www.sciencedirect.com/science/article/B6V0D-4BRPSXB-1/2/ec7f8e6555044c96c75cfef5d9200727>
- [Zhai et Kristensson, 2003] Zhai, S. et Kristensson, P.O. (2003). Shorthand writing on stylus keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, April 05 - 10, 2003). CHI '03. ACM, New York, NY, 97-104. DOI= <http://doi.acm.org/10.1145/642611.642630>
- [Zhai et Kristensson, 2006] Zhai, S. et Kristensson, P.O. (2006). Introduction to Shape Writing, *IBM Research Report RJ10393* (A0611-006), November 1, 2006 , Chapitre de [MacKenzie et Tanaka-Ishii, 2007]
- [Zhai et Kristensson, 2008] Zhai, S. et Kristensson, P.O. (2008). Interlaced QWERTY: accommodating ease of visual search and input flexibility in shape writing. In *Proceeding of the Twenty-Sixth*

- Annual SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy, April 05 - 10, 2008). CHI'08. ACM, New York, NY, 593-596. DOI=<http://doi.acm.org/10.1145/1357054.1357149>
- [Zhang, 1998] Zhang, S.X. (1998). *A High-Performance Soft Keyboard for Mobile Systems*. Master's Thesis. UMI Order Number: UML Order No. GAXMQ-31876., Carleton University. Avalable at : <http://proquest.umi.com/pqdlink?did=734141211&Fmt=7&clientId=79356&RQT=309&VName=PQD>
- [Zribi et Poupée-Fontaine, 2007] Zribi, G. et Poupée-Fontaine, D. (2007). *Dictionnaire du handicap*. École des Hautes Études en Santé Publique, 6e édition. ISBN-13: 978-2859529321

ANNEXES

ANNEXE 1 : TABLE DES CARACTÈRES EDGEWRITE



Cette table de caractères concerne uniquement le mode alphabétique d'EdgeWrite. Elle est reprise de la thèse de Doctorat de Jacob Wobbrock. Pour les tables des autres modes, voir les annexes de [Wobbrock, 2006].

ANNEXE 2 : TABLES DES CARACTÈRES GLYPH

a	b	c	d	e	f	g	h	i	j
ä	å	ç	ð	é	ê	ë	ì	í	î
k	l	m	n	o	p	q	r	s	t
ü	û	w	x	y	z				
0	1	2	3	4	5	6	7	8	9
o	í	2	3	4	5	6	7	8	9
æ	œ	ç	(	)	-	'			
æ	œ	ç	(	)	-	'			

■ 1ère primitive
■ 2ème primitive
■ 3ème primitive

Glyph : Table des caractères alphanumériques.

«	»	‘	’	%	‰	_	/	\	~
«	»	‘	’	‰	‰	_	/	\	~
←	→	↑	↓	↔	↕			•	^
←	→	↑	↓	↔	↕			•	^
\$	€	£	¥	@	©	®	™	§	&
\$	€	£	¥	@	©	®	™	§	&
(	)	[	]	{	}				
(	)	[	]	{	}				

tab

Glyph : Table des caractères spéciaux.

+	-	*	/	^	√	±	°	Σ	Π
+	-	*	/	^	√	±	°	Σ	Π
<	>	≤	≥	=	≠	≈	≡	∅	∞
<	>	≤	≥	=	≠	≈	≡	∅	∞
(	)	[	]	{	}	∇	∃	∈	∉
(	)	[	]	{	}	∇	∃	∈	∉
∩	∪	⊂	⊃	⊆	⊇	⊄	⊅	⊆	⊇
∩	∪	⊂	⊃	⊆	⊇	⊄	⊅	⊆	⊇

Glyph : Table des caractères mathématiques.

0	1	2	3	4
0	1	2	3	4
5	6	7	8	9
5	6	7	8	9
+	-	/	*	=
+	-	/	*	=

Glyph2PPC : Table des caractères

a	b	c	d	e	f	g	h	i	j
ä	å	ç	ð	é	ê	ë	ì	í	î
k	l	m	n	o	p	q	r	s	t
ü	û	w	x	y	z	esp	'	"	
ü	û	w	x	y	z	esp	'	"	

Glyph2PPC : Table des caractères alphabétiques.

à	â	ä	è
à	â	ä	è
é	ê	ë	î
é	ê	ë	î
ï	ô	ö	ù
ï	ô	ö	ù

Glyph2PPC : accentuation des lettres.

+	-	*	/	^	√	(	)
+	-	*	/	^	√	(	)
<	>	≤	≥	=	≠	≈	≡
<	>	≤	≥	=	≠	≈	≡
[	]	±	°	∇	∃	∈	∉
[	]	±	°	∇	∃	∈	∉
∩	∪	⊂	⊃	⊆	⊇	⊄	⊅
∩	∪	⊂	⊃	⊆	⊇	⊄	⊅

Glyph2PPC : Table des caractères mathématiques.

α	β	γ	δ	ε	ζ	η	θ
α	β	γ	δ	ε	ζ	η	θ
ι	κ	λ	μ	ν	ξ	ο	π
ι	κ	λ	μ	ν	ξ	ο	π
ρ	ς	σ	τ	υ	φ	χ	ψ
ρ	ς	σ	τ	υ	φ	χ	ψ
ω							
ω							

Glyph2PPC : Table des lettres grecs.

«	»	‘	’	{	}	%
«	»	‘	’	{	}	%
←	→	↑	↓	↔	↕	
←	→	↑	↓	↔	↕	
	/	\	\$	&	€	©
	/	\	\$	&	€	©
®		@		™	○	~
®		@		™	○	~
§	œ	æ				
§	œ	æ				

Glyph2PPC : Table des caractères spéciaux.

ANNEXE 3 : RÉSULTATS STATISTIQUES DES DIFFÉRENTES COMBINAISONS DE PRIMITIVES SUR LES TOUCHES.

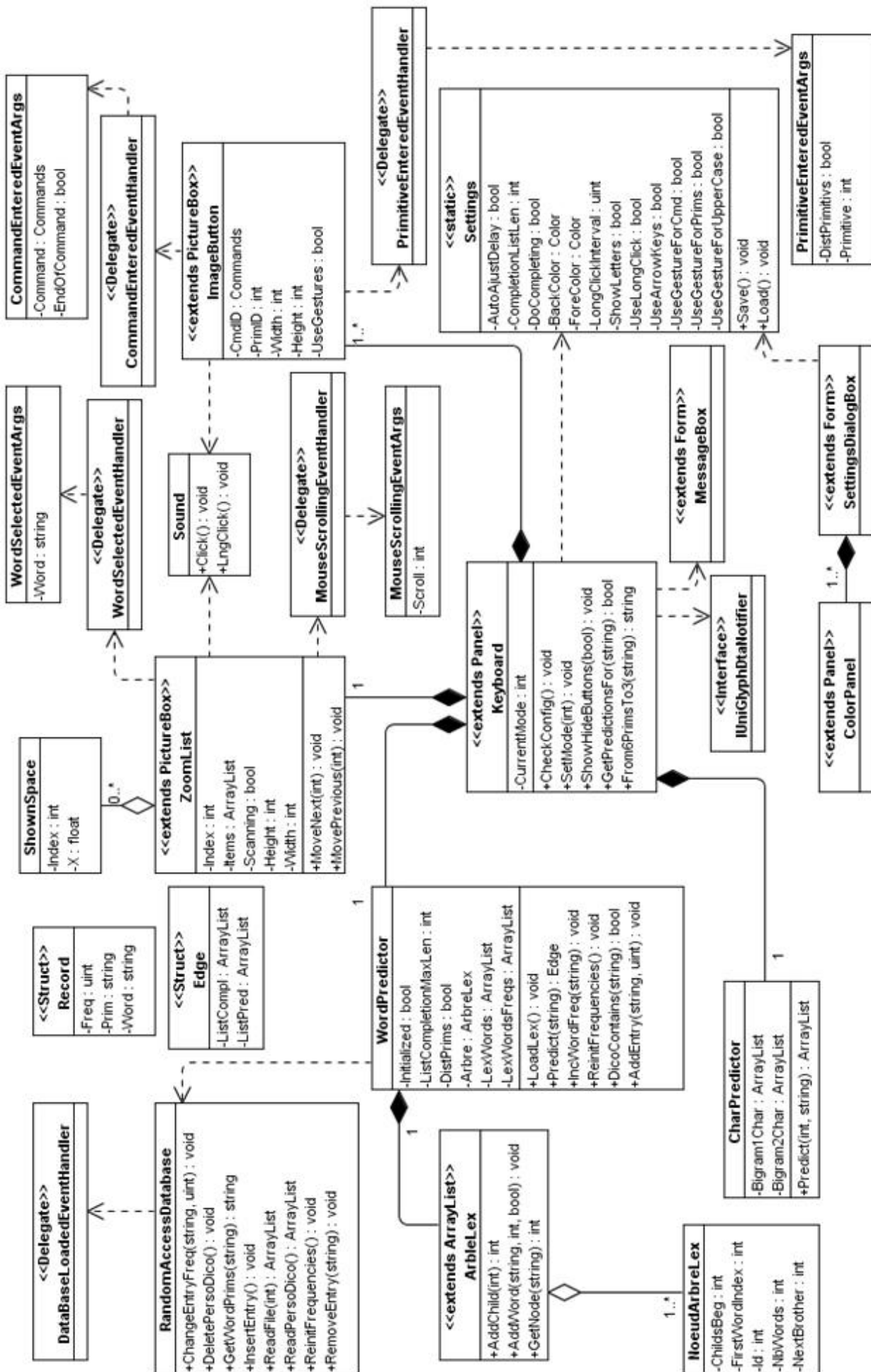
Analogie minuscules	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	'	-
	C	∩	C	C	C	∩	C	∩		∩	∩	∩	∩	∩	C	∩	C		C		∩	∩	∩	∩	∩	∩		-
Analogie majuscules	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	'	-
	/	∩	C	∩			C				/		\	\	C	∩	C	\	C	-		\	\	\	\	/	/	-

Analogie majuscules			
Touches			MCS ¹
/\	-	C∩	2,5202098
/\	-C	∩	3,8967533
/\	-∩	C	2,5543172
/-	\	C∩	3,4837332
/-	\C	∩	3,5888455
/-	\∩	C	2,7868474
/	\-	C∩	3,1475468
/	\C	-∩	3,2586424
/	\∩	-C	3,836645
/C	\-	∩	3,7574525
/C	\	-∩	3,7300034
/C	\∩	-	3,0154245
/∩	\-	C	2,5275126
/∩	\	-C	3,9231646
/∩	\C	-	2,6131985

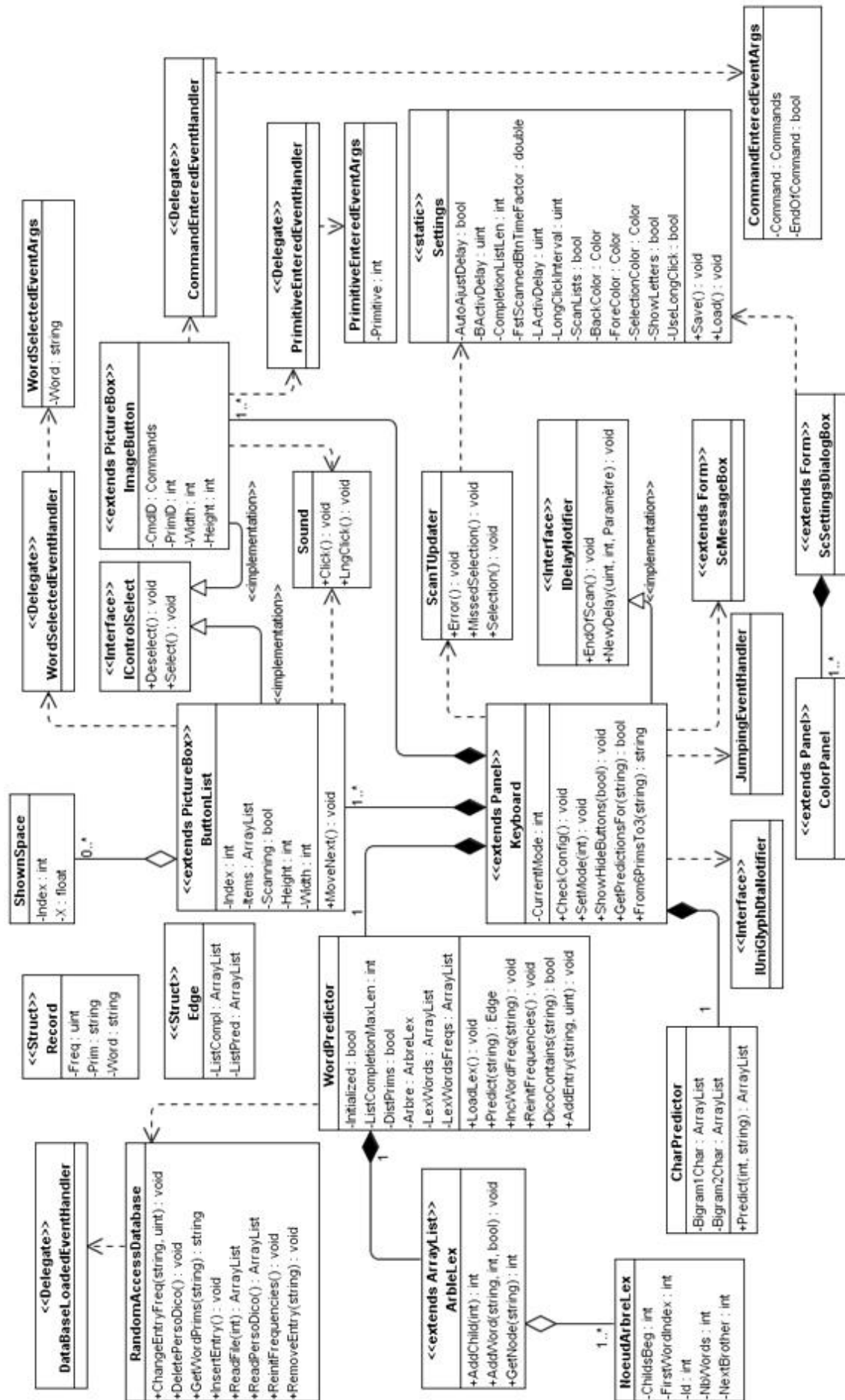
Analogie minuscules			
Touches			MCS ¹
C∩	-	∩∩	3,8783443
C∩	-∩	∩	4,577645
C∩	-∩	∩	5,4809093
C-	∩	∩∩	3,4422033
C-	∩∩	∩	3,248655
C-	∩∩	∩	3,5960376
C	∩-	∩∩	8,195883
C	∩∩	-∩	8,796555
C	∩∩	-∩	8,182268
C∩	∩-	∩	4,2921677
C∩	∩	-∩	4,7850814
C∩	∩∩	-	3,6217525
C∩	∩-	∩	4,6104097
C∩	∩	-∩	4,3402963
C∩	∩∩	-	3,4841197

¹ Nombre Moyen de Candidats par Séquence d'appuis de touches, calculé à partir d'un dictionnaire de 250 000 mots (www.cnam.com/abu).

ANNEXE 4 : STRUCTURE GLOBALE DE UNIGLYPHLIB



ANNEXE 5 : STRUCTURE GLOBALE DE HANDIGLYPHLIB



ANNEXE 6 : LISTE DES PHRASES UTILISÉES DANS LES ÉVALUATIONS

il ne vient au bureau que le mercredi
il pense que tu as toujours raison
ta nouvelle voisine est vraiment belle
vous sortez plus souvent le dimanche
il se tenait immobile dans la cour
il y a un bruit infernal dans la rue
vous pouvez venir chercher vos clefs
il ne pourra pas remporter la victoire
il avait toujours un moment pour dormir
la terre tremblait quand elle criait
il a pris les affaires de sa copine
je suis le moins rapide sur ce circuit
tu peux tout faire seul si tu insistes
la vache regarde passer les trains
la terre tourne autour du soleil
le chien me regarde en remuant la queue
je vais pouvoir refaire toute la maison
il est content de cette trouvaille
tu peux passer des vacances tranquilles
il doit y avoir une explication logique
elles croient toujours aux miracles
il aime aller courir dans les bois
on ne peut pas discuter dans le noir
qui aime la mer aime aussi les bateaux
les louanges sont le protocole des sots
on parle mal quand on ne veut rien dire
les vices empoisonnent les plaisirs
tout usage finit par se changer en abus
le pouvoir sans abus perd son charme
chacun de mes actes est une destruction
que de choses il faut ignorer pour agir
la poule a pondu un œuf au poulailler
une baleine a fait un saut spectaculaire
elle ne peut pas deviner mon origine
les filles de cette villes sont grandes
elle a pris ses jouets et son ours
ils dessinent des oiseaux rouges
elles lisent des belles histoires
la petite fille joue dans la cour
je trouve un pot de confiture de fraise
elle prend une grande enveloppe
il a le nez rouge comme un clown
il ne veut pas partir avec moi
elle pointe un doigt dans sa direction
tu veux bien acheter un kilo de sucre

il aime naviguer sur son voilier
je prends un panier pour les courses
le cowboy joue du banjo au saloon
la voiture est sortie de la route
les phares guident les navires la nuit
le docteur exige que vous vous reposiez
ils sortent plus souvent le dimanche
vous aimez entendre le murmure du vent
la porte donne directement sur la cour
mon voisin de chambre est gentil
le soleil est le fourneau des pauvres
on a toujours plus de bien que de vie
fou qui a le choix et prend le pire
est assez riche qui ne doit rien
les enfants et les fous sont devins
tous les fous sont dans les asiles
les loups ne se mangent pas entre eux
il ne faut pas hurler avec les loups
a quelque chose malheur est bon
les mots ne se battent pas sur le papier
il faut garder une poire pour la soif
la pomme ne tombe pas loin de la branche
mesure la profondeur avant de plonger
les conseillers ne sont pas les payeurs
la roulette russe est un jeu dangereux
on se lasse de tout sauf de comprendre
le bonheur est le plaisir des sages
la raison est ce qui effraie chez un fou
les griffes du chat grattent le tapis
la voile se gonfle dans le vent
le clown a fait rire les enfants
le monde est le plus beau spectacle
tout le monde a son grain de folie
le hasard gouverne nos actions
la jeunesse est une ivresse continuelle
le chemin du devoir est toujours proche
notre tombeau est le cœur des hommes
sans amour les fautes sont grandes
une grande douleur nous rend plus forts
il est absurde de ne pas vouloir changer
il est toujours content de ses cadeaux
ma tante veut que je parte avec elle
la souris danse quand le chat est loin
les chiens ne font pas des chats
une hirondelle ne fait pas le printemps

la belle plume fait le bel oiseau
plus on est de fous plus on rit
il vaut mieux tenir que courir
les bons comptes font les bons amis
les conseillers ne sont pas les payeurs
le soleil luit pour tout le monde
chassez le naturel il revient au galop

un malheur ne vient jamais seul
il ne faut pas hurler avec les loups
la nuit tous les chats sont gris
chien qui aboie ne mord jamais
il faut que la jeunesse se passe
les voyages forment la jeunesse
il faut laver son linge sale en famille

Evaluation de méthodes de saisie

Vous êtes invités à participer à une évaluation de trois méthodes de saisie de texte sur un assistant numérique. Cette évaluation se déroulera sur plusieurs sessions. Pour chaque session, vous aurez à saisir, en utilisant chacune des trois méthodes, cinq phrases choisies aléatoirement.

Pour chaque méthode, vous devrez lire attentivement une fiche descriptive expliquant le principe de la saisie. Vous pourrez ensuite essayer librement de saisir quelques mots et effectuer des commandes de base (suppression, espacement et retour à la ligne). La durée maximale des essais est limitée à 4 minutes.

Enfin, vous passerez à la recopie des phrases qui sera chronométrée et enregistrée automatiquement par l'application. **Les phrases ne contiennent que des lettres minuscules non accentuées et des espaces.** Une seule phrase est affichée à la fois. **La validation d'une phrase pour passer à la suivante se fait par un retour à la ligne.** Pour cette évaluation vous ne devrez saisir que des **lettres**, **l'espacement**, la commande de **suppression** et le **retour à la ligne**.

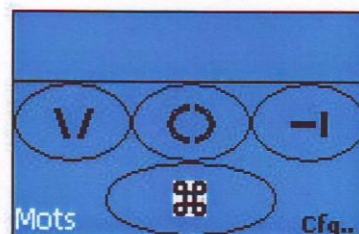
UniGlyph

La saisie de texte avec 4 touches

UniGlyph est une méthode qui permet de saisir du texte avec un clavier de seulement 4 touches en appuyant sur une seule touche pour chaque caractère.

Le clavier UniGlyph se compose des 4 touches suivantes :

- 1 touche de commande "⌘"
- 3 touches dites de forme :
 - o la touche des traits inclinés "∕" (touche 1)
 - o la touche des traits arrondis "()" (touche 2)
 - o la touche des traits droits "—|" (touche 3)



Pour saisir une lettre, il suffit de suivre la règle suivante :

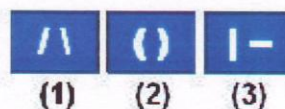
- Si la lettre en majuscule contient un trait incliné alors appuyer sur la touche des "traits inclinés" (1)
- Sinon, si elle contient une boucle alors appuyer sur la touche des "traits arrondis" (2)
- Sinon appuyer sur la touche des "traits droits".

La figure ci-contre montre en rouge, la primitive qui permet de saisir chaque lettre.

A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z

Par exemple :

- "A" est entré en appuyant sur "∕"
- "C" est entré en appuyant sur "()"
- "E" est entré en appuyant sur "—|"



A chaque clic, la zone d'affichage en haut du clavier affiche la liste des mots correspondant à la séquence déjà tapée. On peut faire défiler cette liste avec un geste horizontal sur cette zone. Un simple clic sur le mot désiré permet de le saisir.

Phraze-It

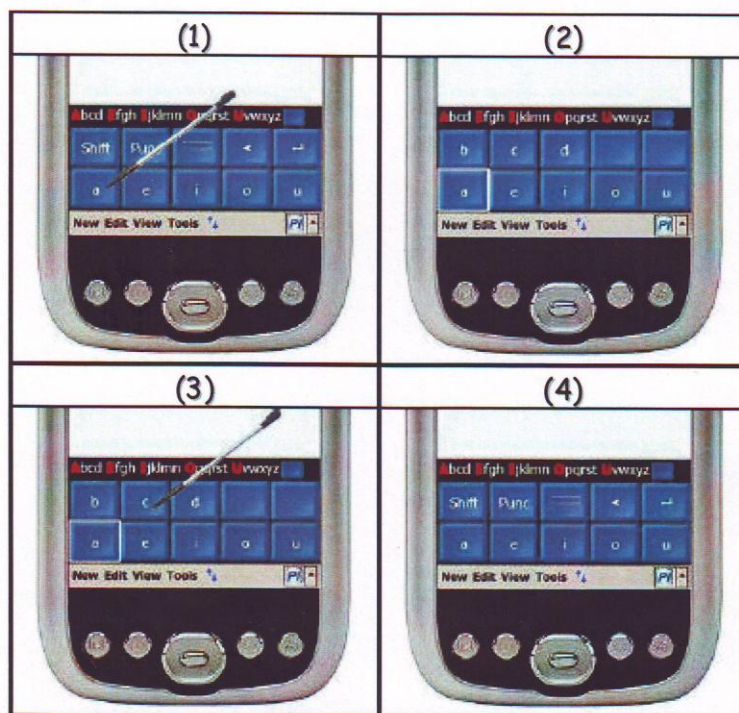
Le clavier Phraze-It permet de saisir les lettres en deux clics.

Le clavier est composé de deux rangées de 5 touches chacune. La première rangée (en bas) contient des voyelles (respectivement : a, e, i, o, u), la deuxième est composée de touches de fonctions (respectivement : Shift, Ponctuation, espace, suppression, retour à la ligne).

Un clic sur une touche de fonction permet d'effectuer la fonction correspondante.

Un clic sur une touche de voyelle permet d'afficher des consonnes sur les touches de la 2^{ème} rangée. Les consonnes affichées sont celles qui suivent la voyelle cliquée dans l'ordre de l'alphabet.

L'exemple ci-dessous, montre la saisie de la lettre "c".



Pour saisir une consonne, il faut effectuer un premier clic sur la voyelle qui la précède directement, puis un deuxième clic sur la consonne désirée.

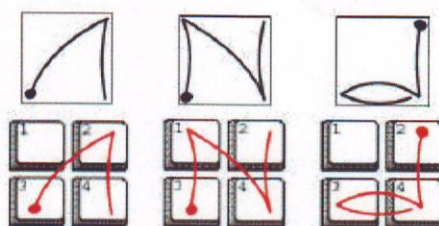
Pour saisir une voyelle, il faut cliquer deux fois sur la touche de la voyelle désirée.

Four-Key EdgeWrite

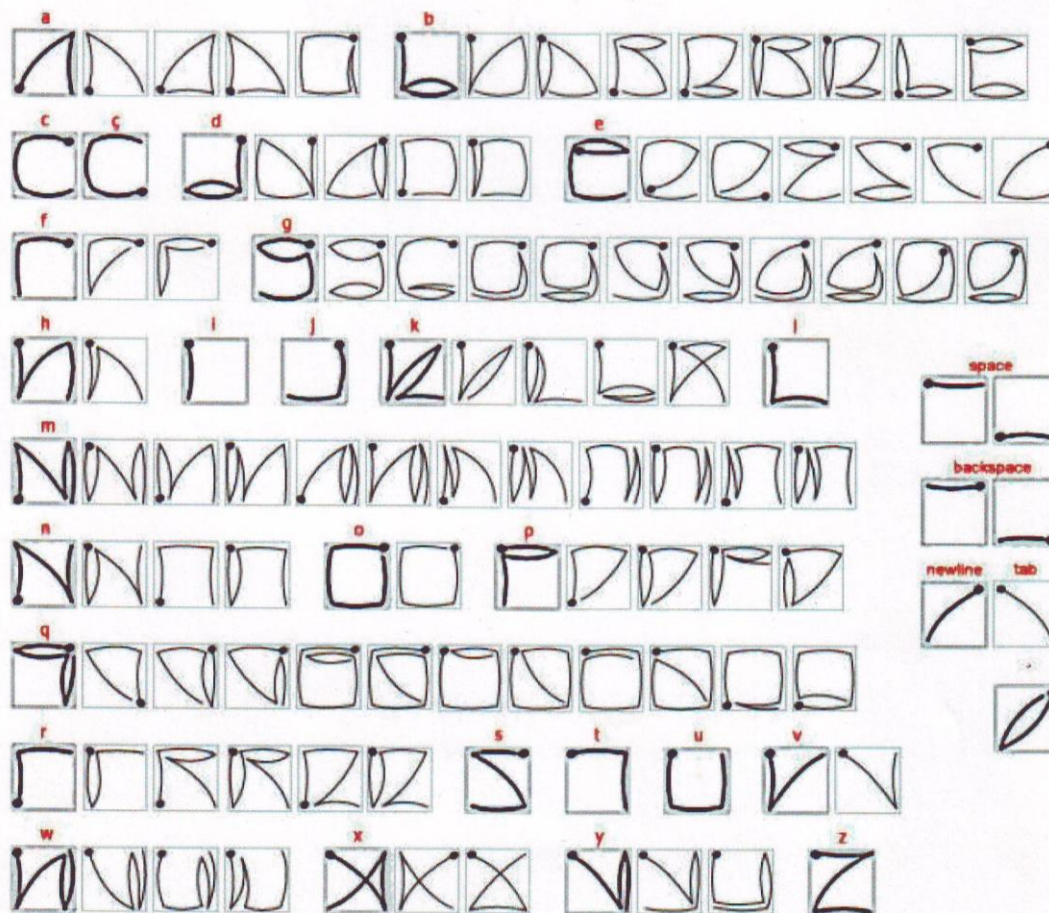
Le clavier Four-Key EdgeWrite est composé de quatre touches.

Pour saisir un caractère, il faut effectuer une séquence de clics sur ces touches de façon à reproduire la forme du caractère désiré.

Par exemple, pour saisir la lettre "A", l'utilisateur doit cliquer, dans l'ordre, sur les touches (3), (2) et (4).



La carte des caractères ci-dessous montre, pour chaque caractère, toutes les représentations possibles. La représentation de base pour chaque caractère est dessinée en gras. Le point marque le début du trait et correspond au premier clic pour saisir chaque caractère.

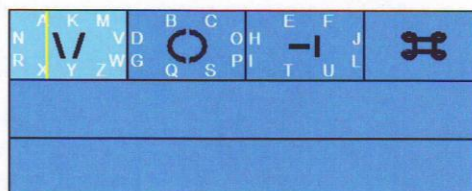


HandiGlyph

HandiGlyph permet aux personnes handicapées de saisir du texte en un seul clic par caractère avec un clavier à défilement de seulement 4 touches en utilisant un simple bouton poussoir.

Le clavier HandiGlyph se compose des 4 touches suivantes :

- 3 touches dites de forme :
 - o la touche des traits inclinés ' \ '
 - o la touche des traits arrondis ' () '
 - o la touche des traits droits ' —| '
- 1 touche de commande ' ⌘ '

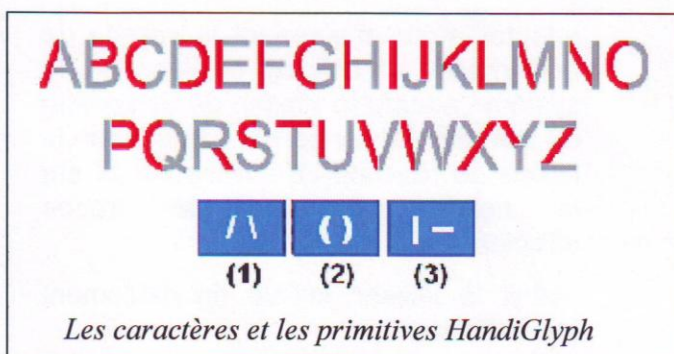


Le clavier HandiGlyph est composé de 4 touches et de deux zones d'affichage.

Pour choisir une touche, il suffit d'attendre qu'elle soit activée (le curseur jaune de défilement passe dessus) et appuyer sur le bouton poussoir.

La saisie se fait par mot. Pour saisir un mot, il faut cliquer sur les touches représentant chacune de ses lettres. Le choix des touches à cliquer se fait selon la règle suivante :

- Si la lettre s'écrit en majuscule avec un trait incliné alors appuyez sur la touche des 'traits inclinés'
- Sinon, si elle s'écrit avec un trait arrondi alors appuyez sur la touche des 'traits arrondis'
- Sinon appuyez sur la touche des 'traits droits'.



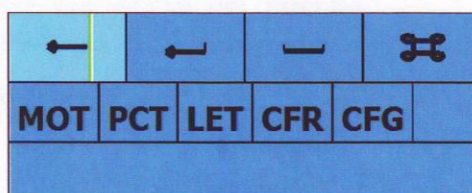
En fait, pour saisir une lettre, il suffit d'appuyer sur la touche qui correspond au trait rouge dans la figure ci-dessus.

Par exemple :

- « A » est saisie avec la touche ' \ '
- « B » est saisie avec la touche ' () '
- « E » est saisie avec la touche ' —| '

A chaque clic, la liste des mots correspondants aux clics effectués s'affiche dans la zone du milieu du clavier. Une autre liste de mots plus longs s'affiche sur la deuxième zone d'affichage du bas. Un appui long ou un appui très long sur l'une des touches de formes permet de basculer respectivement sur la zone du milieu ou celle du bas pour pouvoir sélectionner ensuite le mot désiré par un simple clic. Un appui long sur l'une de ces deux zones d'affichage permet de revenir sur les touches de forme sans sélectionner un mot.

Pour effacer un caractère, insérer un espacement ou un retour à la ligne, il faut effectuer un simple clic sur la touche de commande. Les touches de forme deviennent alors des touches qui permettent de réaliser ces différentes fonctions.



Enfin, avec un clic long sur la touche de commande, les touches de formes permettent d'accéder aux fonctions de suppression d'un mot, d'insertion d'un point ou d'une virgule. Quand une touche est sélectionnée par un clic court, elle est entourée d'un cadre vert, un cadre jaune indique un clic long et un cadre rouge indique un clic très long.

La touche de commande permet aussi d'accéder aux différents modes du clavier HandiGlyph, les abréviations des noms des différents modes s'affichent sur la zone du milieu du clavier :

- MOT : pour le mode de saisie des mots.
- PCT : pour le mode de saisie de ponctuation.
- LET : pour le mode lettres, qui permet de saisir des lettres ou des mots qui n'existent pas dans le dictionnaire. Dans ce cas, après avoir inséré un espace à la fin du mot saisi, un message s'affiche demandant la confirmation de l'ajout du nouveau mot.
- CFR : pour le mode de saisie de chiffres. Chaque chiffre correspond à une ou deux primitives comme indiqué sur la figure ci-contre :
- CFG : pour accéder à l'interface de personnalisation du clavier. Cette interface permet de :

0123456789

- adapter automatiquement la vitesse de défilement : en activant cette option, le système adapte la vitesse de défilement du curseur jaune en se basant sur le temps de réaction de l'utilisateur et sur le nombre d'erreurs de frappe effectuées.
- définir la vitesse initiale du défilement (en millisecondes).
- définir le temps de sélection de la première touche. Ce temps doit être supérieur ou égale à la période du défilement.
- scanner toujours la zone de prédiction : par défaut, après le défilement sur la dernière touche (de commande), le curseur de défilement revient à la première touche de primitives (inclinés). Pour passer à la zone d'affichage des prédictions, l'utilisateur doit effectuer un clic long sur l'une des touches de primitives. Dans le cas où cette option est activée, le curseur passe toujours par la zone d'affichage avant de reprendre son cycle en commençant par la première touche de primitives. Cela évite les clics longs mais ralentit considérablement la saisie.
- afficher les lettres sur les touches : cette option permet d'afficher sur chaque touche, autour des deux primitives, toutes les lettres correspondantes.
- choisir les couleurs du clavier.

☒ Adapter automatiquement la vitesse.
Période de défilement initiale : 1000ms

- +

Temps de pause sur la première touche : 1500ms

- +

☒ Scanner tjrs la zone de prédiction.
Durée du long clic : 1000ms

- +

☒ Afficher les lettres sur les touches.
Couleurs de base :

A A A A

Couleur de selection :

Fermer