



Systèmes de saisie de texte et techniques de pointage pour des contextes d'interaction complexes

Mathieu Raynal

► To cite this version:

Mathieu Raynal. Systèmes de saisie de texte et techniques de pointage pour des contextes d'interaction complexes. Interface homme-machine [cs.HC]. Université toulouse 3 Paul Sabatier, 2021. <tel-03727912>

HAL Id: tel-03727912

<https://hal.science/tel-03727912v1>

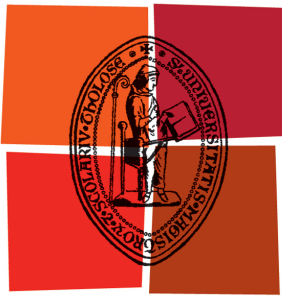
Submitted on 19 Jul 2022

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.



HAL Authorization



Université
de Toulouse

Mémoire

en vue de l'obtention de l'Habilitation à Diriger des Recherches (HDR)

Délivrée par :

Université Toulouse III Paul Sabatier (UT3)

Discipline ou spécialité :

Informatique - Interaction Humain-Machine

Présentée et soutenue par :

Mathieu RAYNAL

le 25 novembre 2021

Titre :

Systèmes de saisie de texte et techniques de pointage
pour des contextes d'interaction complexes

École doctorale :

Mathématiques Informatique Télécommunications (MITT)

Unité de recherche :

IRIT - UMR 5505

Jury :

Isabelle Pecci, rapporteuse, MCF HDR - Université de Lorraine

Jean-Yves Antoine, rapporteur, Prof., Université de Tours

Franck Poirier, rapporteur, Prof, Université de Bretagne Sud

Véronique Gaildrat, examinatrice, Prof., Université Toulouse III

Christophe Jouffrais, examinateur, DR CNRS, IPAL-CNRS, Singapour

Table des matières

Table des matières	iii
Liste des tableaux	iv
Liste des figures	v
Introduction	1
I Systèmes de saisie de texte	4
Introduction	6
1 Aide à la conception de claviers logiciels	9
A Espace de description DESTIT	10
1 Les critères de description	10
2 Représentation des claviers logiciels dans notre espace de description	14
B Aide à la conception des claviers logiciels	17
1 Architecture logicielle	18
2 Le langage de description KeySpec	20
C Synthèse et perspectives	21
2 Modèle et outil pour l'évaluation théorique	23
A Étude du comportement de l'utilisateur lors d'une tâche de saisie	24
B Évaluation des performances sur un clavier avec une liste de mots	27
C Synthèse et perspectives	30
3 Intégration des résultats de la prédiction	32
A Intégration de la prédiction de mots	32
B Intégration de la prédiction de caractères	36
C Utilisation de la déduction de mots	43
D Synthèse	46

II Dispositifs et techniques de pointage	48
Introduction	50
4 Aide à la conception et à l'évaluation de TI	53
A Évaluation de la tâche de pointage : Fuzzy Fitts	53
B Évaluation du pointage pour les environnements 3D	57
1 Adaptation de la norme d'évaluation	58
2 Plate-forme d'évaluation pour les pointages en 3D	60
C 3 DIM – Modèle de description des dispositifs de pointage 3D	62
D Synthèse	65
5 Conception de TI pour des environnements « complexes »	67
A Pointage sur une grande surface d'affichage distante	67
1 Manipulation d'un dispositif isométrique logiciel	68
2 Techniques de pointage reposant sur l'élasticité du poignet	69
B Interaction avec un environnement virtuel 3D	72
1 Environnement virtuel 3D affiché sur une surface distante	72
2 Environnement 3D affiché sur Smartphone	74
C Pointage dans un environnement hétérogène	76
D Synthèse	78
III Synthèse et perspectives	81
Synthèse	83
Perspectives	87
Bibliographie	93

Liste des tableaux

1	Résultats des évaluations comparant le système WordTree et la liste de prédiction de mots	34
2	Résultats des évaluations comparant le clavier Centralist et le clavier alpha . .	36
3	Synthèse de mes contributions	85

Liste des figures

1	Description du système FOCL qui utilise un système de prédiction de caractères	15
2	à gauche, le système Sybille [WANDMACHER et al., 2008], et sa description dans notre espace à droite	16
3	à gauche, codage des caractères avec EdgeWrite, à droite, description de EdgeWrite, pour la version sur PDA avec le calque [WOBBROCK et al., 2003]	16
4	Clavier multitap conçu dans [VIGOUROUX et al., 2004] et sa description dans DESTIT	17
5	Architecture logicielle, basée sur l'utilisation de 5 modules, pour construire tout type de claviers logiciels	19
6	Temps de saisie d'un caractère en fonction de la difficulté pour atteindre la touche correspondante	25
7	Principaux résultats sur les fixations du regard sur la liste de mots.	25
8	Sélection dans la liste en fonction du nombre de caractères déjà saisis	26
9	Principaux résultats obtenus lors de l'étude du comportement de l'utilisateur au cours d'une tâche de saisie	26
10	Interface de prototypage et d'évaluation de clavier logiciel et liste de prédiction	30
11	WordTree après la saisie du préfixe "bo"	33
12	Disposition du clavier CentraList	35
13	Clavier pour la version 2 des KeyGlasses : A. clavier simple; B. KeyGlasses qui apparaissent après la saisie du 's'; C. KeyGlasses qui apparaissent après la saisie du 'e' qui était placé sur une KeyGlass. Extrait de [RAYNAL, 2014]	38
14	Disposition des touches sur Floodkey après la saisie de « joue »	39
15	Disposition des caractères sur le clavier Spreadkey	40
16	Vitesse de saisie de texte (en caractères par seconde) au cours des 10 premières sessions avec le simple clavier QWERTY et augmenté du système SpreadKey . .	41
17	Vitesse de saisie de texte (en mots par minute) en fonction du nombre de phrases saisies et du clavier utilisé	42
18	Vitesse de saisie avec les deux claviers à l'issue de la première et de la deuxième session	45
19	Vitesse de saisie avec les deux claviers à l'issue de la deuxième session	46

20	Vitesse de saisie de texte (en mots par minute) en fonction du nombre de phrases saisies et du clavier utilisé	55
21	régression linéaire classique; régression floue linéaire; régression floue non linéaire	55
22	Vitesse de saisie avec les deux claviers	58
23	Répartition des cibles et exemple de tâche de pointage	60
24	Plateforme pour les tâches de pointage en 3D	61
25	Lien entre modèle de tâches et graphe de scène	63
26	Les 6 blocs d'un lien entre une sous-tâche et un élément du graphe de scène . .	63
27	Illustration du fonctionnement du Softstick (à gauche), et de son utilisation sur une interface (à droite)	69
28	Étapes d'une tâche de pointage avec la technique hybride lorsque $D > D_1$. (a) Le curseur se déplace sur une distance D_1 , qui correspond à un mouvement d_1 le long de la plage isotonique de l'entrée. (b) La fenêtre se déplace vers la cible, qui apparaît à une distance D_2 du centre initial de la fenêtre. La vitesse v de la fenêtre augmente au fur et à mesure que la position de l'entrée s'étend vers ses côtés extrêmes. (c) Le curseur se déplace vers la cible en couvrant une distance D_3	71
29	Contrôle de position non-uniforme. (a) Pointage absolu dans la zone centrale de la plage d'entrée avec un faible gain C-D. (b-d) Recalibrage de la position avec des mouvements de va-et-vient	71
30	Résultats	72
31	Les 3 techniques	73
32	Exemple de sélection dans un ensemble d'éléments pré-sélectionnés : la cible à atteindre est le cube représenté en violet sur l'image de gauche	75
33	Exemple de sélection dans un ensemble d'éléments pré-sélectionnés : la cible à atteindre est le cube représenté en violet sur l'image de gauche	77

Introduction

D'après le Centre de Recherche pour l'ÉtuDe et l'Observation des Conditions de vie (Crédoc)¹, le taux d'équipement des ménages français en ordinateur de bureau atteint, depuis maintenant 4 ans, plus de 80%, alors qu'il n'était que de 34% au début des années 2000. Outre les ordinateurs de bureau, d'autres dispositifs émergent. Par exemple, l'utilisation de Smartphones et de tablettes est en pleine explosion : en l'espace de 4 ans, le taux d'équipement de ces deux types de dispositifs est passé respectivement de 17% et 4% en 2011 à 58% et 35% en 2015. Ces petits dispositifs, au départ utilisés de manière ludique (jeux, internet, réseaux sociaux, etc.), sont devenus au fil du temps de véritables outils de travail, notamment pour les tâches de bureautique (messaging, édition de texte, etc.). Ils sont maintenant en train de s'imposer dans de nombreux métiers comme un nouvel outil de travail avec des applications dédiées permettant notamment des tâches qui sont régulièrement réalisées en mobilité.

Cette démocratisation de l'informatique grand public entraîne une évolution de l'usage du numérique. Par exemple, l'état français s'est lancé, depuis quelques années, dans un processus de dématérialisation des services publics : il s'agit ainsi de remplacer tous les formulaires et autres documents administratifs papier par une version numérique. Cependant, pour que la transition vers du tout numérique soit possible, il est important que chacun puisse utiliser un outil informatique quel que soit son contexte d'utilisation. Cela nécessite de prendre en compte la diversité des utilisateurs et de permettre l'accessibilité à tous de l'outil informatique, quelles que soient les éventuelles déficiences motrices ou sensorielles. Par exemple, une personne souffrant d'une déficience motrice des membres supérieurs ne pourra pas forcément utiliser les dispositifs traditionnels (généralement le clavier et la souris). Il est donc nécessaire de proposer des dispositifs physiques et/ou logiciels adaptés aux capacités motrices et sensorielles de chaque personne, de manière à ce qu'elle puisse accéder aux outils informatiques [VIGOUROUX et al., 2005 ; ORIOLA et al., 2007].

1. http://www.arcep.fr/uploads/tx_gspublication/CREDOC – Rapport – enquete – diffusion – TIC – France_CGE – ARCEP_nov2015.pdf

D'autre part, le fait d'enregistrer et stocker de plus en plus d'informations sous la forme numérique engendre de nouvelles problématiques : le nombre de données a explosé et les types de données collectées sont de plus en plus variés. De ce fait, les interfaces WIMP, en deux dimensions, classiquement utilisées au début des années 2000, ne sont plus adaptées pour pouvoir analyser de grandes quantités de données hétérogènes. Des outils spécialisés pour les analyser sont proposés, en utilisant par exemple des interfaces en 3D, parfois combinées à des interfaces en 2D. Malheureusement, ces interfaces, si elles répondent à une problématique de visualisation et de recherche de données, restent encore généralement utilisées avec le traditionnel couple clavier / souris, alors qu'il existe de nombreux dispositifs offrant plus de possibilités d'interaction à l'utilisateur. De ce fait, l'interaction avec ces nouveaux outils n'est généralement pas adéquate et les tâches sont réalisées de manière non optimisée.

Mes travaux de recherche se sont focalisés sur deux des principaux piliers de l'interaction : d'une part les systèmes de saisie de texte qui permettent aux utilisateurs d'entrer des informations, et d'autre part, les techniques de pointage qui permettent aux utilisateurs de sélectionner un élément dans un environnement virtuel. La sélection est la première étape indispensable pour consulter, modifier, déplacer l'élément en question. De façon plus spécifique, mes travaux de recherche ont vocation à contribuer à ces deux enjeux majeurs : 1) l'accessibilité du numérique pour tous, et plus particulièrement l'accès à la saisie de texte pour les personnes souffrant d'une déficience les empêchant d'utiliser les moyens de saisie proposés au grand public ; et 2) la manipulation de données dans des environnements virtuels complexes (grandes quantités d'information, environnements virtuels 3D, données hétérogènes). Ma problématique de recherche se situe donc dans le domaine de l'interaction Homme-Machine et concerne principalement l'étude et l'optimisation des performances obtenues au moyen de techniques de pointage ou de saisie de texte. Les performances d'un système se caractérisent à la fois par la vitesse d'exécution d'une tâche et d'autre part par sa faculté à la réaliser de manière précise (sans erreur).

Ce mémoire est structuré autour de ces deux piliers de l'interaction : la première partie est consacrée à mes travaux réalisés sur la saisie de texte, où je me suis plus particulièrement intéressé à l'intégration des systèmes de prédiction au sein des claviers logiciels. Pour ces travaux, le domaine d'application privilégié était l'utilisation de ces systèmes de saisie de texte par des personnes ayant une déficience motrice. La seconde partie présente mes travaux effectués sur le pointage. Mes travaux avaient pour objet l'étude des dispositifs de pointage utilisés pour la sélection d'objets dans des environnements complexes : grand espace d'affichage, environnement 3D, environnement hétérogène (2D + 3D).

Dans les deux cas, la méthodologie de travail a été identique : 1) des travaux fondamentaux qui visent d'une part à modéliser l'espace de conception du domaine étudié, et d'autre part à proposer des méthodes d'évaluation pertinentes pour les contextes d'interaction étudiés ; 2)

des travaux appliqués consistant à concevoir et évaluer des claviers logiciels ou des dispositifs de pointage en vue d'optimiser les performances de saisie ou de pointage.

Enfin, après une synthèse de mes travaux passés, je terminerai ce mémoire par la proposition d'un ensemble d'axes de recherche que je compte approfondir pour la suite de mes recherches.

Première partie

Systèmes de saisie de texte

Introduction

Dans cette partie, nous nous intéressons plus particulièrement à la saisie de texte effectuée au moyen d'un système de saisie numérique, c'est-à-dire tout système logiciel qui permet de saisir des informations sans avoir recours à un clavier physique. Les premiers systèmes de ce type étaient les claviers dits logiciels. Ces derniers étaient des représentations à l'identique des claviers physiques : la disposition des caractères était similaire. Chaque touche était représentée par un bouton et l'utilisateur pouvait ainsi interagir avec ce clavier avec son dispositif de pointage. Ces claviers ont été créés à l'origine pour permettre aux personnes avec déficiences des membres supérieurs de saisir du texte sans avoir à utiliser un clavier physique. Ils pouvaient alors utiliser les claviers logiciels avec n'importe quel type de dispositif de pointage (souris traditionnelle, trackball, joystick, système de suivi du regard, etc.). Ces dernières années, avec l'émergence des Smartphones et tablettes, ces systèmes de saisie de texte se sont démocratisés du fait de la disparition des claviers physiques sur ces nouveaux dispositifs. Les nouveaux périphériques d'interaction tels que les montres connectées, les manettes de jeux, ou encore les casques de réalité virtuelle, incitent également à utiliser des systèmes de saisie de texte logiciels pour permettre la saisie de données avec ces dispositifs. De plus, du fait que ces systèmes soient logiciels, ils offrent beaucoup plus de possibilités en termes d'interaction qu'un clavier physique. Ainsi, ces systèmes sont capables d'évoluer dynamiquement au cours de la saisie de l'utilisateur, d'interagir avec d'autres composants logiciels, ou d'utiliser plusieurs modalités d'interaction.

Toutes ces possibilités ont ainsi donné lieu à une grande variété de systèmes de saisie de texte logiciels. Cependant, la vitesse de saisie sur ce type de clavier reste inférieure aux vitesses de saisie que l'on peut avoir avec un clavier physique. C'est pourquoi la problématique principale reste la même quel que soit le contexte d'utilisation : améliorer les performances de saisie de texte de ces dispositifs.

Actuellement, il n'existe pas d'espace de conception permettant de situer ces travaux par rapport à l'ensemble des systèmes proposés. Les études se limitent généralement à des com-

paraisons avec des systèmes proposés dans le contexte d'interaction étudié (par exemple, saisie sur ordinateur par des personnes ne pouvant utiliser de dispositif de pointage, saisie sur smartphone par des non-voyants, ou encore saisie sur montres connectées). Or, des solutions proposées pour un contexte donné pourraient être réutilisées ou, du moins, adaptées à un autre contexte. C'est pourquoi, dans le premier chapitre de cette partie, je proposerai un espace de description des systèmes de saisie de texte afin de pouvoir décrire chaque système selon une liste de critères que je définirai. Cet espace, nommé DESTIT, nous permettra ainsi d'avoir une vue d'ensemble des différents systèmes de saisie de texte proposés dans la littérature et de les classer selon un ou plusieurs critères. Ce chapitre terminera par la présentation de notre architecture logicielle KeySpec. L'architecture modulaire que nous proposons a pour enjeu de permettre la modélisation de tout type de système de saisie de texte quel que soit son contexte d'utilisation.

Les claviers logiciels sont généralement associés à un système de prédiction de mots. Ceci permet ainsi à l'utilisateur d'avoir moins d'actions à effectuer pour saisir un mot. En effet, tout au long de la saisie de l'utilisateur, le système de prédiction propose les mots les plus probables en fonction des caractères saisis. Ces mots sont alors disposés à côté du clavier pour que l'utilisateur puisse les sélectionner plutôt que de saisir le reste des caractères composant le mot. Cependant ces systèmes de prédiction n'apportent pas toujours une amélioration dans les performances de saisie. Dans le deuxième chapitre, je présenterai notre étude sur l'utilisation des listes de mots proposées à proximité du clavier logiciel. L'objectif était d'analyser pourquoi les utilisateurs perdent du temps en utilisant une liste de prédiction. Pour mener à bien cette étude, nous nous sommes appuyés sur l'analyse du suivi du regard de l'utilisateur au cours de la saisie. Je présenterai également le modèle prédictif qui découle des résultats de cette étude, et l'outil que nous avons conçu et développé pour simuler les performances de saisie d'un clavier logiciel couplé à une liste de mots prédits.

Enfin, dans le troisième et dernier chapitre de cette partie, je présenterai les différents claviers logiciels couplés à un système de prédiction de mots ou de caractères que nous avons proposés. Ce type de système a pour but de proposer les mots ou caractères qui ont le plus de chance d'être saisis par l'utilisateur. L'intérêt de l'intégration d'un tel système est de faciliter la sélection des mots ou caractères prédits pour que l'utilisateur puisse les sélectionner plus rapidement. Nos travaux sur les listes de mots prédits tentent de répondre aux problèmes soulevés lors notre étude présentée au chapitre 2. Dans le cas de l'utilisation de la prédiction de caractères, la prédiction apporte en permanence des changements sur le clavier. Ces modifications peuvent perturber l'utilisateur en apportant de nouvelles informations sur le clavier, ou en modifiant la structure de ce dernier. Nous présenterons donc les différentes stratégies que nous avons mises en place pour intégrer les résultats de la prédiction de caractères tout en minimisant les perturbations engendrées pour l'utilisateur. Enfin, je présenterai un dernier clavier que nous avons réalisé dans un contexte différent : le clavier DUCK, qui est

un clavier que nous avons conçu pour une utilisation sur smartphone par des personnes déficientes visuel. Ce clavier fait également intervenir un corpus linguistique, mais avec une approche différente basée sur la déduction du mot à partir d'un ensemble de tache sur le clavier logiciel.

Chapitre 1

Aide à la conception de claviers logiciels

Comme nous l’avons vu précédemment, par rapport à un clavier physique classique, un système de saisie de texte logiciel offre de nombreuses possibilités d’interactions supplémentaires, permet l’utilisation d’algorithmes de prédiction, déduction ou correction et peut être couplé avec des composants logiciels complémentaires telles que des listes et des menus. Plusieurs revues de la littérature ont été effectuées sur la saisie de texte [DUBE et ARIF, 2019; LUNA et al., 2018; MACKENZIE et R. W. SOUKOREFF, 2002; MARTIN et PECCI, 2007; POIRIER et al., 2018; SIQUEIRA et al., 2016]. Cependant, vu la diversité des systèmes, des contextes d’interaction, ou encore des modalités d’interaction employées, celles-ci prennent le parti de présenter les systèmes en fonction d’un ou plusieurs critères précis (accessibilité en fonction d’une déficience motrice [POIRIER et al., 2018] ou visuelle [SIQUEIRA et al., 2016], usage en mobilité [MACKENZIE et R. W. SOUKOREFF, 2002], utilisation dans un environnement virtuel 3D [DUBE et ARIF, 2019], etc.) et de comparer les systèmes en fonction de ce prisme.

Dans un premier temps, je présenterai dans ce chapitre notre espace de description DESTIT (pour « DDescription Space for Text Input Techniques ») conçu de manière à décrire tous les claviers logiciels quels que soient le contexte et les contraintes pour lesquels ils ont été conçus. L’objectif est de pouvoir représenter les différents types de claviers logiciels existants selon les éléments qui les caractérisent et donc quels que soient le contexte d’utilisation, les modalités d’interactions utilisées et les composants logiciels couplés au clavier.

Puis, je présenterai l’architecture logicielle que nous avons développée pour faciliter la conception de systèmes de saisie de texte. Basé sur le principe de conception modulaire, notre souhait était de proposer un ensemble de briques logicielles entièrement paramétrables et interchangeables qui seraient utilisables par tout concepteur de système de saisie de texte. Notre objectif est ainsi d’uniformiser la conception des systèmes de saisie de texte pour faciliter l’évaluation et la comparaison de ces systèmes.

A Espace de description DESTIT

L'enjeu de notre espace de description est de pouvoir décrire chaque clavier logiciel d'un point de vue système : c'est-à-dire représenter la manière dont l'utilisateur va pouvoir interagir avec le clavier, décrire le fonctionnement interne de ce dernier et notamment comment celui-ci va évoluer au fur et à mesure de la saisie et enfin indiquer la nature des éléments produits par le clavier logiciel (caractères, chaînes de caractères, mots ...). Cependant, au-delà des aspects purement fonctionnels, nous contextualisons également chaque clavier logiciel en décrivant l'environnement pour lequel ce clavier a été conçu.

Cet espace de description ne concerne que les claviers logiciels. Nous définissons par claviers logiciels tout système de saisie de texte composé de plusieurs zones interactives permettant de produire des chaînes de caractères. Sur un clavier logiciel, les zones interactives sont généralement matérialisées par des boutons auxquels sont liés un ou plusieurs caractères ou codes permettant d'aboutir à un caractère par une séquence d'interactions. Nous ne prenons pas en compte ici les systèmes ayant recours à des systèmes de reconnaissance de geste ou du signal. Ainsi, nous ne traitons pas des systèmes de reconnaissance d'écriture ou encore des systèmes de dictée vocale qui permettent également de saisir du texte.

1 Les critères de description

Notre espace de description est défini selon 4 dimensions. La première concerne le contexte d'utilisation du clavier logiciel : les caractéristiques de l'utilisateur et de l'environnement pour lequel le système a été conçu. La deuxième dimension décrit la représentation du clavier logiciel, c'est-à-dire la manière dont est organisé le clavier et sa représentation graphique. La troisième s'intéresse à la manière dont l'utilisateur va interagir avec le système de saisie, à la fois pour naviguer sur le clavier et pour sélectionner les éléments souhaités. Enfin, la dernière dimension définit le système linguistique utilisé pour faciliter la saisie avec ce clavier logiciel. Chacune de ces dimensions est décomposée en critères que je présente succinctement ici.

Contexte d'utilisation

Les claviers logiciels ont été initialement conçus pour les personnes qui ne peuvent utiliser un clavier physique et ainsi leur permettre de saisir du texte sur leur ordinateur. Cependant, depuis l'émergence des Smartphones et plus généralement des écrans tactiles, le clavier logiciel est devenu un outil incontournable pour la plupart d'entre nous. La conception d'un clavier logiciel répond généralement à une problématique bien précise, souvent liée soit aux capacités de l'utilisateur, soit aux caractéristiques du dispositif sur lequel ce clavier sera utilisé. Le contexte d'utilisation est donc décomposé en trois critères :

Les utilisateurs visés – Il s’agit d’indiquer si ce clavier est conçu pour un profil de personnes en particulier. Il peut s’agir de capacités motrices ou sensorielles de l’utilisateur au moment de devoir utiliser ce clavier, mais également de l’expertise de l’utilisateur.

La surface d’affichage – Si les premiers systèmes de saisie de texte étaient principalement prévus pour une utilisation sur un ordinateur de bureau traditionnel où le clavier logiciel était affiché sur l’écran, les usages se sont largement diversifiés depuis une vingtaine d’années. La diversité des surfaces d’affichage (Smartphones, Smartwatch, TV interactive, Hololens, etc.) montre l’importance de ce critère au moment de concevoir un système de saisie de texte. La disposition du clavier ne pourra pas être la même s’il s’agit d’un clavier affiché sur l’écran d’une montre ou si celui-ci est présenté dans un casque de réalité mixte. Une petite surface rendra plus compliquée la représentation du système.

Le dispositif d’interaction – Au-delà de la surface d’affichage, l’autre critère important lors de la conception du système réside dans les modalités d’interaction à disposition de l’utilisateur pour interagir avec le clavier logiciel. Une modalité d’interaction telle que définie par [NIGAY, 1997] est décrite par un couple dispositif physique / langage d’interaction. Ce critère permet de déterminer quel est le dispositif physique (souris, joystick, Wii, écran tactile, etc.) utilisé pour interagir avec le clavier logiciel. Comme nous le verrons sur la dimension « interaction », la saisie se décompose généralement en deux principales tâches : d’une part la navigation et d’autre part la sélection. Le langage d’interaction utilisé peut être différent pour ces deux tâches. C’est pourquoi le langage d’interaction utilisé sera décrit dans les critères de « navigation » et « validation » de la dimension « interaction ».

Représentation

Nous verrons lors de la représentation de notre espace de description qu’un clavier logiciel peut parfois être décomposé en plusieurs systèmes de saisie de texte. Par exemple, nous considérons un clavier logiciel couplé à une liste de mots comme une combinaison de deux systèmes de saisie : le clavier d’une part et la liste de mots d’autre part. Les valeurs que nous affectons à chacun des critères que nous définissons par la suite seront attribuées séparément à chaque partie de notre clavier logiciel.

La deuxième dimension de notre espace de description concerne la structure et la représentation graphique de notre clavier logiciel. Nous utilisons 4 critères pour décrire la représentation de chaque partie de notre clavier logiciel.

La visibilité – Certaines parties du clavier ne sont pas visibles en permanence. Par exemple, le système POBox [MASUI, 1999] fait apparaître une liste des mots les plus probables une fois le premier caractère du mot saisi. Nous distinguons donc les parties qui sont visibles en permanence de celles qui le sont ponctuellement.

Le langage – Le langage détermine l’ensemble des informations liées aux zones interactives. Cet ensemble peut être l’ensemble des caractères présents et saisissables sur le système, un ensemble non défini de mots (comme pour les listes de mots prédits) ou un ensemble de codes permettant de définir les différents éléments d’un code. Par exemple, EdgeWrite [WOBBROCK, 2006] utilise 4 codes représentant les 4 coins d’un carré. Tous les caractères sont alors obtenus par une séquence de ces codes.

La cardinalité – La cardinalité correspond au nombre de zones interactives présentes sur le système de saisie. Nous notons « complet » lorsqu’il y a autant de zones interactives que d’éléments possibles dans l’ensemble décrivant le langage. Dans le cas où il y a moins de zones que d’éléments, le nombre de zones est directement donné. On parle alors de clavier ambigu. Plusieurs éléments du langage seront associés à une même zone, et c’est l’interaction utilisée sur la zone ou l’algorithme de déduction qui permettra de choisir l’élément à sélectionner sur la zone interactive.

La disposition – Nous décrivons ici comment les différentes zones interactives sont disposées les unes par rapport aux autres. Cela peut être une grille quelconque telle que la disposition AZERTY, une matrice avec un nombre fini de lignes et de colonnes, une liste horizontale ou verticale, une disposition circulaire, ou encore sous forme de carré.

Interaction

Quel que soit le type d’éléments produits (codes, caractères ou mots), la production de ceux-ci se décompose en deux étapes : dans un premier temps la sélection de la zone interactive souhaitée puis sa validation.

La navigation – La phase de navigation sur un clavier logiciel ne sera présente que pour les systèmes où l’on utilise un pointeur ou curseur sur le clavier pour sélectionner les zones interactives souhaitées. Pour les claviers logiciels présents sur les smartphones, montres connectées ou tout autre dispositif utilisant un écran tactile, il n’y a pas de pointeur à déplacer, l’utilisateur pointe directement avec ses doigts ou un stylet sur les zones. Dans ce cas, nous considérons qu’il n’y a pas de phase de navigation. Si un pointeur ou curseur est présent, le déplacement de celui-ci dépend des capacités de l’utilisateur à manipuler un dispositif de pointage. Pour les utilisateurs ayant une déficience motrice sévère, il est impossible d’utiliser un dispositif de pointage. La navigation se fait alors au moyen d’un curseur qui se déplace automatiquement de zone en zone. Lorsque le curseur est sur la bonne zone, l’utilisateur n’a plus qu’à la valider. Si l’utilisateur peut utiliser un dispositif de pointage, il déplace alors un pointeur qu’il manipule au moyen du dispositif de pointage. Il existe deux types de déplacement : d’une part, le déplacement continu d’un pointeur en utilisant un dispositif de pointage et d’autre part, un déplacement de zone en zone (dit « discret ») d’un curseur au moyen de touches directionnelles ou autres actions discrètes permettant de matérialiser les

différents sens de déplacement. Nous distinguons donc 3 types de navigation du pointeur : automatique, continu, discret.

La validation – Enfin, la validation d’une zone interactive permet de produire l’élément qui y est lié. Pour l’utilisation des claviers logiciels sur Smartphones ou autre écran tactile, la validation la plus employée consiste à réaliser une tape directement sur la zone souhaitée. L’appui du doigt sur l’écran valide ainsi la zone située sous le doigt. Lors d’une interaction par tracé, les zones interactives sont généralement validées par franchissement : chaque zone traversée par le tracé est alors considérée comme validée. Il est également possible de réaliser un geste sur la zone souhaitée pour valider cette dernière. Cette technique de validation par geste est généralement utilisée pour une zone sur laquelle se trouvent plusieurs caractères. L’utilisation d’un petit ensemble de gestes pour valider la zone permet en même temps de lever l’ambiguïté sur le caractère souhaité. Pour les claviers logiciels utilisés au moyen d’un dispositif de pointage, la validation se fait généralement grâce à un bouton sur le dispositif de pointage. De même, pour les claviers dont la sélection de la zone s’effectue par défilement automatique, l’utilisateur valide la sélection en utilisant un contacteur ou toute autre modalité d’interaction lui permettant d’effectuer une action discrète pour indiquer le moment où il souhaite valider la sélection [MACKENZIE et ASHTIANI, 2011].

Système linguistique

Enfin, la dernière dimension de notre espace de description détermine si notre clavier logiciel utilise des algorithmes se basant sur des connaissances linguistiques. Ce type d’algorithme se base sur les actions précédemment effectuées par l’utilisateur et sur un ensemble de connaissances linguistiques (règles, statistiques, etc.) et permet soit de compléter/modifier une chaîne de caractères déjà saisie, soit de modifier, dynamiquement au cours de la saisie, une partie du clavier pour aider l’utilisateur dans sa saisie de texte.

Ces systèmes peuvent être de différents types. Les plus connus sont les algorithmes de prédiction de mots qui vont proposer l’ensemble des mots les plus probables en fonction du préfixe saisi. Les systèmes de prédiction peuvent également proposer un ensemble de caractères correspondant aux caractères qui ont le plus de chance de succéder au préfixe déjà saisi. Les claviers logiciels peuvent également faire appel à un système de déduction. Ces systèmes utilisent un ensemble d’informations provenant de la saisie de l’utilisateur pour en déduire le mot qu’il a souhaité saisir. Ces informations peuvent être de différentes natures, par exemple, la séquence ordonnée des zones que l’utilisateur a survolé ou validé, ou encore une séquence de coordonnées correspondant aux positions du doigt ou du pointeur au moment de chaque tape ou clic sur le clavier logiciel. A partir de ces informations et de ces connaissances linguistiques, le système va alors en déduire le mot qui correspond le mieux à ces informations.

Dans le cas où plusieurs mots auraient des correspondances équivalentes, le système peut également proposer un ensemble de mots.

2 Représentation des claviers logiciels dans notre espace de description

Cet espace de description se présente sous la forme d'un tableau (voir les exemples des figures 1 2, 3 et 4) avec un critère par ligne. Au-delà de l'aspect descriptif de cet espace, l'objectif est également de montrer la dynamique du clavier logiciel et les interactions possibles entre les différentes parties du clavier. En premier lieu, les cercles verts représentent les productions possibles : les caractères 'C', 'M' ou 'S' affichés sur ces cercles signifient que le système produit des caractères, des mots ou des chaînes de caractères. Les flèches bleues représentent les interactions entre les différents critères du clavier logiciel : le rond bleu indique le critère qui engendre l'interaction alors que le losange bleu indique le critère impacté par cette interaction. Les rectangles verts et bleus définissent respectivement les informations qui sont envoyées au système linguistique, et la modification qui est apportée sur le clavier logiciel en réaction aux informations envoyées par le système linguistique. Par exemple, le clavier F.O.C.L. est un clavier logiciel conçu à la fin des années 90 pour être affiché sur un téléphone portable [BELLMAN et MACKENZIE, 1998]. L'utilisateur déplaçait alors un curseur de touche en touche au moyen des 4 touches directionnelles du téléphone. Après chaque caractère saisi par l'utilisateur, la disposition des caractères est modifiée de manière à ce que les caractères les plus probables soient le plus près possible du curseur. La Figure 1 nous donne la description de ce clavier dans notre espace. Nous pouvons observer que les caractères saisis sont ajoutés au texte produit (cercle vert avec un 'C') et envoyés parallèlement au système de prédiction de caractères qui lui donne en retour la liste des caractères ordonnés en fonction de la probabilité de chacun d'être saisi. Grâce à cette information, le système met à jour sa disposition de caractères.

Si le clavier logiciel est composé de plusieurs composants logiciels permettant de saisir de différentes manières des éléments, ceux-ci seront présentés en colonne les uns à côté des autres (cf. Figure 2). Par exemple, un clavier logiciel accompagné d'une liste de mots sera considéré comme un système composé de 2 composants logiciels : d'une part le clavier logiciel et d'autre part la liste des mots prédits. La bordure utilisée entre les deux types de composant sert à décrire la relation entre ces deux parties. Les deux systèmes peuvent être utilisés en parallèle (bordure en pointillés). L'utilisateur peut, par exemple, passer du clavier à la liste de prédiction et inversement au cours de la saisie d'un mot. Les deux systèmes sont utilisés de manière disjointe et séquentiellement (bordure pleine). Par exemple, sur le système DUCK [RAYNAL et ROUSSILLE, 2017], l'utilisateur effectue la saisie approximative de l'ensemble des caractères composant le mot avant de sélectionner le bon mot dans la liste des mots proposés par le système de déduction.

Contexte	Utilisateur	Tous
	Affichage	Écran téléphone
	Dispositif	Touches directionnelles du téléphone
Clavier	Visibilité	Permanent
	Langage	{caractères}
	Cardinalité	Complet
	Disposition	Grille 
Interaction	Navigation	Appui touches directionnelles
	Validation	Appui touche 
Syst. Linguistique	Prédiction de caractères	

FIGURE 1 – Description du système FOCL qui utilise un système de prédiction de caractères

Par exemple, le système Sybille [WANDMACHER et al., 2008] est un clavier logiciel conçu pour les personnes handicapées des membres supérieurs et qui ne peuvent utiliser de dispositif de pointage. La navigation sur le clavier logiciel se fait alors par défilement automatique. L'utilisateur valide la sélection à l'aide d'un contacteur qu'il active au moment où le curseur se trouve sur la touche souhaitée. Sybille est un système de saisie de texte très complet qui propose l'ensemble des caractères et fonctionnalités proposés sur un clavier standard, mais également des touches complémentaires qui proposent des fonctionnalités supplémentaires (cf. Figure 2). Il a la particularité d'être composé de plusieurs composants de saisie (un pour la saisie des caractères, une liste de mots, un pavé numérique, et une barre de fonctionnalités). De même, 2 systèmes de prédictions linguistiques sont utilisés de manière complémentaire : d'une part, un système de prédiction de caractères qui permet de ré-agencer les caractères après chaque saisie d'un caractère par l'utilisateur et, d'autre part, un système de prédiction de mots qui permet de proposer les mots les plus probables à l'utilisateur. La représentation de Sybille dans notre espace de description est présentée à la Figure 2. Nous ne présentons ici que la partie de Sybille qui évolue dynamiquement au cours de la saisie : i.e. le bloc de saisie de caractères dont la disposition des caractères est réévaluée après chaque caractère saisie, et la liste de mots qui est également mise à jour après chaque nouveau caractère.

Pour une grande partie des claviers logiciels, chaque validation d'une zone par l'utilisateur provoque la production d'un caractère ou d'un code de son dictionnaire. Dans le cas des claviers à base de code, une séquence de codes est réalisée avant d'être envoyée à un algorithme de déduction qui détermine le caractère correspondant à la séquence. Cette séquence est produite en validant plusieurs zones interactives. Pour matérialiser cette séquence dans notre espace de description, nous utilisons une flèche arrondie noire entre le critère de sé-

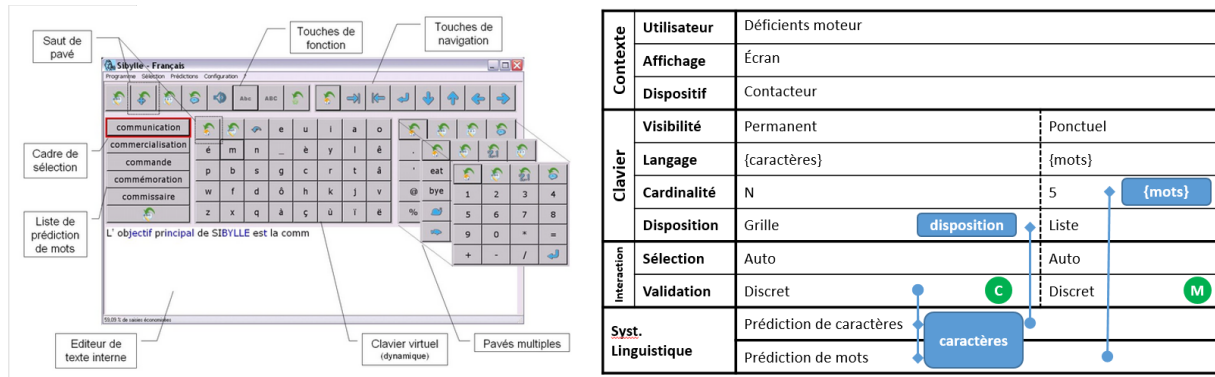


FIGURE 2 – à gauche, le système Sybille [WANDMACHER et al., 2008], et sa description dans notre espace à droite

lection et le critère de validation. Le nombre de répétition nécessaire pour aboutir à la séquence est donné sous forme d'intervalle à proximité des flèches. Par exemple, dans EdgeWrite [WOBBROCK et al., 2003], chaque caractère est codé par une séquence de coins du carré par lesquels l'utilisateur est passé (cf. Figure 3 - gauche). La partie droite de la Figure 3 présente la description de EdgeWrite avec notre espace de description.

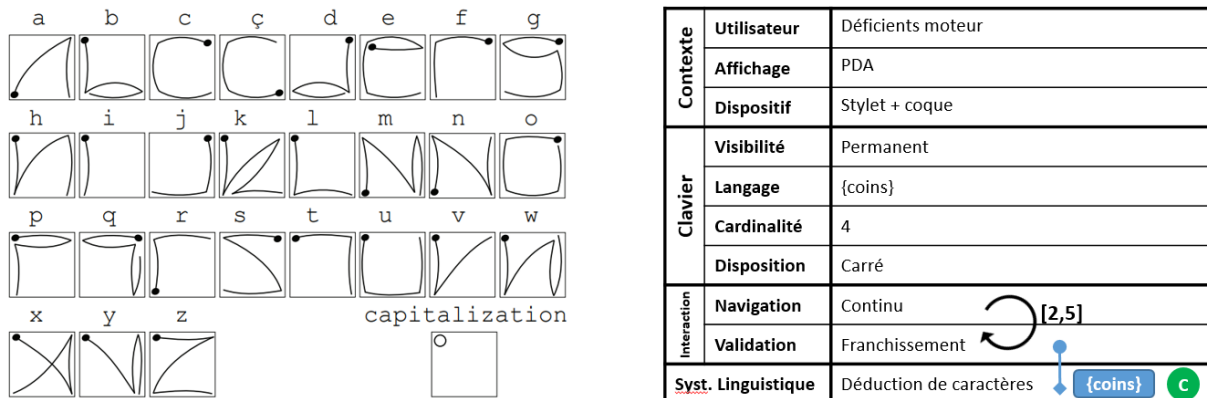


FIGURE 3 – à gauche, codage des caractères avec EdgeWrite, à droite, description de EdgeWrite, pour la version sur PDA avec le calque [WOBBROCK et al., 2003]

Enfin, la flèche arrondie peut également être utilisée seulement sur le critère de validation pour représenter une répétition de la phase de validation. Ce type de répétition est notamment utilisé pour lever une ambiguïté lorsque plusieurs caractères se trouvent sur la même zone interactive. Par exemple, la sélection d'un caractère sur un clavier logiciel multitap s'effectue au moyen d'un, deux ou trois clic(s) sur le bouton contenant le caractère souhaité [VIGOUROUX et al., 2004]. La figure 4 représente le fonctionnement du clavier multitap avec notre espace de description

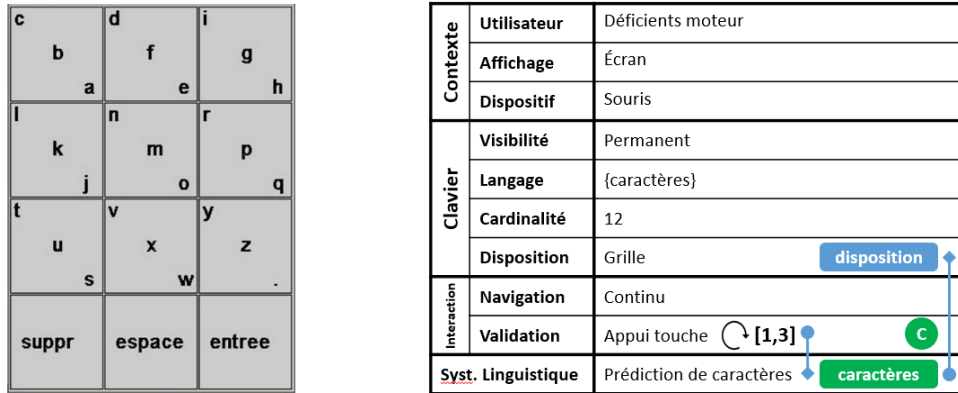


FIGURE 4 – Clavier multitap conçu dans [VIGOUROUX et al., 2004] et sa description dans DESTIT

B Aide à la conception des claviers logiciels

Si l'espace de description présenté à la section précédente permet d'explorer l'ensemble des solutions proposées, d'analyser et de comparer différents claviers logiciels, cela reste une description très haut niveau et ne permet pas de mettre en évidence les différences de conception entre des claviers logiciels qui se baseraient sur une base apparemment commune. L'exemple le plus répandu concerne la disposition des touches d'un clavier logiciel traditionnel. Au-delà de réelles différences entre deux dispositions (par exemple QWERTY versus AZERTY), il existe souvent des différences entre des dispositions a priori identiques : en effet, chaque concepteur va paramétrer la disposition de son clavier en proposant sa propre disposition des touches avec des positions et des tailles de touches différentes, ou en utilisant un ensemble de caractères pas toujours identique d'un clavier à un autre (certains ne vont utiliser que les 26 caractères de l'alphabet latin et la barre espace quand d'autres vont ajouter les caractères accentués, la ponctuation et/ou les chiffres). Toutes ces différences de conception peuvent compliquer la comparaison de ces différents systèmes.

Lors de ma thèse [RAYNAL, 2005], j'avais conçu et développé la plate-forme e-assiste [RAYNAL et al., 2005] qui avait pour enjeu d'uniformiser les évaluations de claviers logiciels, en proposant un ensemble de composants logiciels utilisables pour concevoir son dispositif d'évaluation, ainsi qu'une librairie JAVA permettant d'encapsuler un système de saisie dans la plate-forme e-assiste. Je proposais également un protocole de communication normalisé que chaque concepteur pouvait utiliser lors de la création de son système de saisie de texte et ainsi utiliser les mêmes outils et les mêmes métriques d'évaluation lors de l'évaluation de son système. Cette plateforme permettait ainsi de normaliser la manière d'évaluer les systèmes de saisie de texte et d'harmoniser les métriques d'évaluation utilisées (vitesse de saisie, taux d'erreur, taux d'utilisation d'un système de prédiction, etc.). En revanche, la plateforme e-

assiste ne proposait pas d'outils de conception ou de briques logicielles pouvant faciliter la conception des systèmes de saisie de texte.

Durant la thèse de Bruno Merlin, que j'ai encadré de juin 2008 à juin 2011, nous avons travaillé sur cette problématique de l'uniformisation des composants de base qui constituent les systèmes de saisie de texte. Pour répondre à cette problématique, nous avons conçu une architecture logicielle permettant de composer un clavier logiciel. Nous sommes partis du constat qu'un clavier logiciel est composé de composants identiques (ensemble de touches, système de prédiction) quel que soit le système de saisie de texte que l'on souhaite réaliser. Nous proposons une architecture logicielle basée sur l'utilisation de 5 modules qui communiquent entre eux (cf. Figure 5). Ces modules sont entièrement paramétrables, et un protocole de communication facilite l'échange de données entre ces différents modules. Ainsi à partir de ces modules, un concepteur peut créer son propre clavier en instanciant ces modules et en les assemblant. A noter qu'une instance d'un module peut être réutilisée sur différents systèmes de saisie de texte. Ceci permet ainsi d'assurer une certaine uniformité entre plusieurs systèmes de saisie de texte. Par exemple, si deux systèmes ne diffèrent que par le système de prédiction qu'ils utilisent, ces systèmes peuvent utiliser les mêmes modules pour le reste de leur fonctionnement et permettre ainsi de focaliser l'évaluation sur le système de prédiction.

1 Architecture logicielle

La Figure 5 présente les 5 modules que nous avons mis en œuvre, ainsi que le type d'information qu'il est possible d'échanger entre modules.

Le module « Interaction Manager » – A la manière de ce que propose ICON [DRAGICEVIC et FEKETE, 2001], le module « Interaction Manager » écoute les différents événements reçus et les transmet aux autres modules en fonction de leur pertinence. Il est également capable d'interpréter une séquence d'événements (par exemple une succession de positions du pointeur du dispositif de pointage) et de générer un événement unique envoyé ensuite au reste du système. Par exemple, dans le cas de MessageEase, l'utilisateur peut soit cliquer sur une touche pour sélectionner le caractère central de la touche, soit utiliser un geste directionnel pour désigner un autre caractère présent sur la touche. Dans tous les cas, le module « Interaction Manager » reçoit un premier événement correspondant à la pression du bouton du dispositif de pointage. Celui-ci sera envoyé au module « Layout manager » pour lui permettre de désigner la touche ciblée. Si cette pression est suivie d'un ensemble de déplacements avant le relâchement du bouton, le module interprétera cette séquence d'événements en un geste directionnel. Le geste reconnu sera à son tour envoyé au « Layout manager » pour lever l'ambiguïté sur le caractère souhaité. Dans le cas où l'utilisateur relâche directement le bouton du

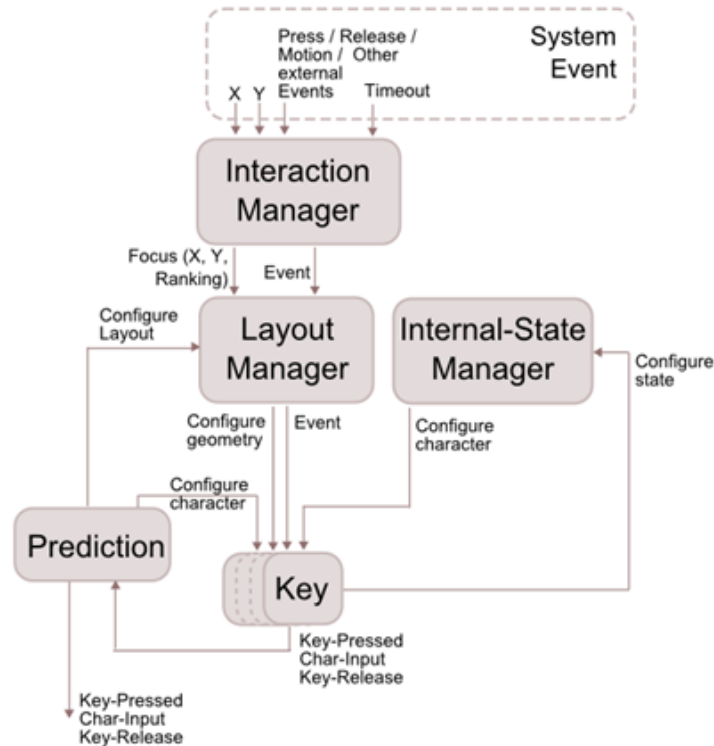


FIGURE 5 – Architecture logicielle, basée sur l'utilisation de 5 modules, pour construire tout type de claviers logiciels

dispositif de pointage sans faire de geste, le module envoie au reste du système un événement clic bouton et non un geste.

Le Module « Layout Manager » organise l'agencement et la géométrie des touches. Pour les claviers dont la disposition est statique, ce module ne fait que disposer les touches sur le clavier. Dans le cas des ré-agencements dynamiques des touches au cours de la saisie pour des claviers tels que Sybille [WANDMACHER et al., 2008] ou F.O.C.L [BELLMAN et MACKENZIE, 1998], ce module gère également le ré-agencement des touches au cours de la saisie. Enfin, dans le cas des claviers qui utilisent le défilement automatique, le « Layout manager » gère également le déplacement du focus d'une touche à une autre.

Le module « Touche » gère le rendu et le fonctionnement des touches. D'une part, il gère les différents aspects graphiques de la touche en fonction de son état (enfoncé, relâché, survolé, désactivé, etc.), et organise également l'affichage des caractères en fonction de l'état de la touche (dans le cas où celle-ci contient plusieurs caractères visibles), et de l'état du système (en fonction des modes activés : Shift, Ctrl, Maj, etc.). L'état du système est géré par le module « Internal State Manager »

Le module « Internal State Manager » détermine, en fonction des touches fonctionnelles (Ctrl, Shift, Alt, Maj) activées, un état interne du clavier. Cet état détermine l'ensemble des

caractères actifs sur le clavier à cet instant. Le changement d'état impacte la distribution des caractères et éventuellement la représentation de la touche. L'état interne peut être modifié par l'appui sur une touche fonctionnelle, mais également par une interaction ou par la pression sur une touche standard. Par exemple, après l'appui sur « Shift », l'appui sur une touche quelconque rétablit l'état initial du clavier et impacte donc l'état interne du clavier. Ce module n'est pas le plus utilisé car il implique que nous ayons un clavier complet qui utilise les touches fonctionnelles, ce qui est rarement le cas dans les prototypes de recherche qui sont réalisés.

Le module « Prédiction » – Comme nous l'avons vu précédemment, les systèmes de prédiction observés reposent sur deux principaux types d'algorithmes : d'une part, les algorithmes de prédiction du caractère suivant en fonction des N précédents caractères saisis, ou prédiction des mots les plus probables en fonction du préfixe saisi; d'autre part, ceux utilisés pour déduire un mot à partir d'une séquence d'événements (il peut s'agir de lever une ambiguïté lorsque plusieurs caractères sont présents sur une même touche [VIGOUROUX et al., 2004], ou encore de déterminer un mot par rapport à un chemin réalisé sur le clavier avec le doigt ou le pointeur du dispositif de pointage [KRISTENSSON et ZHAI, 2004]). Le module « Prédiction » reçoit donc les événements relatifs à la saisie sur le clavier (caractères saisis et donc émis par les modules « Touche » ou plus globalement les événements émis par le module « Interaction Manager » qui permettent de déterminer le mot saisi). Les résultats de l'algorithme sont ensuite propagés par le module « Prediction » aux modules « Layout Manager » et « Touche » pour que ces derniers puissent appliquer les modifications sur le clavier (par exemple, réorganisation des touches sur le clavier, mise à jour de la liste de mots, etc.). Ces résultats peuvent également être envoyés au système d'exploitation pour que le résultat soit ajouté au texte saisi.

2 Le langage de description KeySpec

Afin de pouvoir implémenter les différents types de système de saisie de texte sur la base de cette architecture logicielle, les modules présentés ci-dessus ont été instanciés de manière à être entièrement paramétrables. Pour faciliter le paramétrage de ces modules, nous avons défini le langage de description KeySpec permettant de paramétrer et d'assembler ces différentes entités pour ensuite générer un clavier. Le langage a pour avantage de fournir les éléments essentiels pour élaborer et générer des claviers complexes et ne nécessite pas de connaissances avancées en programmation pour créer et adapter l'essentiel des claviers. Néanmoins, il requiert pour l'instant des connaissances de base sur la structure du XML. Cependant, le caractère descriptif de la conception des claviers via KeySpec permet d'envisager une interface graphique supplantant l'écriture du code XML.

En l'état, le langage nous permet de générer la grande majorité des claviers logiciels existants, sans nécessiter de développements informatiques supplémentaires. C'est le cas par exemple de tous les claviers statiques (GAG [RAYNAL et VIGOUROUX, 2005a; RAYNAL, 2006], Metropolis, Fitaly, AZERTY, etc.), des claviers ambigus désambiguïsés par interaction (MultiTap, Glyph2, Multi-Ring, etc.), d'une grande partie des claviers désambiguïsés par un système de prédiction (T9, LeterWise, etc.), des claviers additionnés de listes de prédiction (claviers standards ou ambigus) et d'une partie des claviers dynamiques (KeyGlass, SibyMot, FOCL, etc.).

Néanmoins, certains claviers évoluant dynamiquement et de façon complexe en fonction des résultats d'un système de prédiction n'échappent pas à des modifications en Java du comportement standard des composants. Ces claviers bénéficient toutefois de toute la structure de base fournie par le langage et concentre ces interventions programmées sur les détails critiques qui ne peuvent être pris en charge par le langage. Par exemple, certains claviers dont les transformations graphiques dynamiques sont complexes tels que VirHKey ou Dasher échappent aux capacités du langage.

C Synthèse et perspectives

Le pouvoir de description d'un espace désigne sa capacité à décrire un ensemble significatif de solutions existantes. Au-delà des systèmes de saisie de texte décrit précédemment, nous avons décrit avec cet espace plus de 25 systèmes de tout type dans différents contextes.

Pour que cet espace de description puisse servir à la communauté travaillant dans le domaine de la saisie de texte, nous avons conçu et développé un site web dédié aux systèmes de saisie de texte : <http://text-entry.com/>¹

Ce site web est en premier lieu une vitrine des différents systèmes de saisie de texte existants et a pour objectif de regrouper un maximum de prototypes de recherche. Au-delà d'un simple catalogue de solutions existantes, le but de notre site web est aussi de permettre d'explorer cet ensemble de systèmes en fonction des caractéristiques souhaitées. Pour cela, notre site web reprend l'ensemble des critères de notre espace de description pour permettre de filtrer l'ensemble des solutions selon ces critères. La description dans notre espace est proposée pour chaque système présenté sur notre page. Ce site continue à être alimenté avec les systèmes de saisie de texte que nous trouvons dans la littérature. D'autre part, une page de contact est disponible sur le site web pour permettre aux concepteurs de nous proposer d'intégrer leurs solutions au site web.

1. Note aux rapporteurs : Au moment de vous envoyer le manuscrit, le site n'est pas totalement terminé ... il sera en ligne le 7 août :)

Cet espace de description est le fruit d'un travail de collaboration avec Scott Mackenzie de l'université de York (Toronto, Canada), et Georges Badr, un de mes anciens doctorants, actuellement enseignant chercheur à l'université Saint Antonin (Liban). Carine HADDAD, co-encadrée par Georges BADR et moi et qui débutera une thèse sur la saisie de texte à la rentrée 2021, a également participé à ce travail de synthèse sur les systèmes de saisie de texte. Enfin le site web proposé a été réalisé au cours de plusieurs stages d'étudiants de Licence Informatique au cours des années 2020 et 2021.

Les travaux sur l'architecture et le langage de description sont présentés plus en détails dans le mémoire de thèse de Bruno Merlin [MERLIN, 2011]. Ces travaux ont été présentés lors du workshop « *Designing and Evaluating Text Entry Methods* » à l'occasion de la conférence CHI 2012 [MERLIN et al., 2012a].

Pour aller au-delà du simple aspect descriptif, notre espace gagnerait à proposer une évaluation théorique des différentes solutions. Cela permettrait de comparer plus aisément des solutions entre elles. Les performances théoriques des systèmes de saisie de texte ont fait l'objet de plusieurs études [DAS et STUERZLINGER, 2008; PAVLOVYCH et STUERZLINGER, 2004; W. R. SOUKOREFF et MACKENZIE, 1995; SEARS et al., 2001; SILFVERBERG et al., 2000; ZHAI et al., 2002]. Cependant, ces modèles répondent à un contexte bien précis, pour des tâches de saisie simples où généralement plusieurs paramètres sont laissés de côté [MERLIN et RAYNAL, 2012]. Notre espace DESTIT décompose le fonctionnement des claviers logiciels, notamment en décrivant le fonctionnement des sous-tâches telles que la navigation au sein du clavier et la validation d'un caractère. A partir de cette décomposition, et des différents critères de notre espace, il apparaît donc possible de choisir plus aisément le modèle théorique le plus approprié au clavier que l'on décrit. D'autre part, plusieurs modèles pourraient même s'avérer complémentaires pour évaluer plus finement certaines caractéristiques du clavier en question. Tel le modèle KLM [CARD et al., 1980], on pourrait combiner l'utilisation de plusieurs modèles théoriques, chacun spécifique à une sous-tâche, pour proposer une évaluation théorique plus précise du clavier. Pour réaliser ce travail, il sera nécessaire d'effectuer une revue systématique de la littérature sur les modèles théoriques proposés pour la saisie de texte, et d'analyser comment associer chaque modèle à un de nos critères ou à une combinaison de critères.

Chapitre 2

Modèle et outil pour l'évaluation théorique des performances de l'utilisateur

Au-delà de la conception de nouveaux systèmes, il est également important de prendre conscience de l'impact que peuvent avoir nos choix de conception. Pour cela, il est possible d'utiliser des modèles théoriques qui permettent de prédire les performances de l'utilisateur en fonction du contexte. Soukoreff et Mackenzie [[W. R. SOUKOREFF et MACKENZIE, 1995](#)] ont par exemple proposé un modèle théorique permettant de prédire les performances d'un utilisateur novice et d'un utilisateur expert sur un clavier statique avec une disposition de caractères donnée.

Sur des claviers évoluant dynamiquement, il est plus difficile d'évaluer théoriquement leur apport du fait des modifications incessantes du système qui peuvent plus ou moins perturber l'utilisateur. J'avais également pu le constater lors de mes travaux de thèse sur le système KeyGlass [[RAYNAL, 2007](#)] où l'écart entre les performances réelles et théoriques était très important.

Ce constat m'a donc amené à m'intéresser aux facteurs qui peuvent perturber l'utilisateur lors de la saisie sur un système de saisie impliquant des modifications tout au long de la saisie. Pour cela, je me suis intéressé plus particulièrement au comportement de l'utilisateur au cours de la saisie sur un clavier logiciel couplé à une liste de prédiction. De plus, pour tenter de comprendre le comportement de l'utilisateur au cours de la saisie, j'ai choisi d'étudier la position du regard de l'utilisateur tout au long de la saisie.

A Étude du comportement de l'utilisateur lors d'une tâche de saisie

Les claviers logiciels couplés à une liste de prédiction sont les systèmes les plus connus et les plus répandus. Ceux-ci proposent les mots les plus probables pour un préfixe donné. Les résultats suggérés par ces systèmes sont alors proposés sous forme de liste à proximité du clavier logiciel. L'intérêt d'utiliser ce type de système est de minimiser le nombre d'actions à réaliser : plutôt que de saisir l'ensemble des caractères du mot, l'utilisateur peut sélectionner le mot dès que celui-ci apparaît dans la liste. Plus le mot apparaît rapidement dans la liste et moins l'utilisateur aura à effectuer d'actions pour saisir un mot. Cependant, ces listes de mots peuvent s'avérer contre-productives si le mot n'apparaît pas dans la liste ou s'il apparaît tardivement. En effet, l'utilisateur doit regarder la liste des mots assez régulièrement pour voir si le mot qu'il souhaite saisir se trouve dans la liste.

Plusieurs travaux ont été effectués sur les listes pour montrer l'importance de la position de la liste à l'écran, la taille de l'espace d'affichage et le nombre d'items dans la liste. Cependant, si ces travaux permettent d'optimiser l'utilisation de la liste en faisant varier ces paramètres, ils n'expliquent pas pourquoi l'utilisation de la liste reste préjudiciable pour l'utilisateur dès que celui-ci a une vitesse de saisie un peu élevée sur le clavier logiciel. C'est pourquoi nous avons choisi d'étudier l'utilisation de la liste de mots.

Pour notre étude, nous avons choisi de prendre une liste de prédiction avec les caractéristiques qui lui sont normalement favorables. Ainsi, notre liste de prédiction est une liste verticale de sept items où les mots sont classés par rapport à leur fréquence d'apparition. Seuls des mots de plus de 6 caractères y sont affichés. Enfin, les mots affichés sont ordonnés par rapport à leur fréquence d'apparition (et non par ordre alphabétique). Notre étude consistait à saisir 40 mots que nous avons choisis en fonction du moment où ils apparaissent dans la liste et de leur position dans la liste au moment où ils apparaissent. Nous avons donc choisi des mots qui apparaissent après 2, 4, 6 et 8 caractères saisis et aux première, troisième, cinquième et septième positions dans la liste. Ainsi, nous avons choisi 2 mots pour chacun de ces couples « nombre de caractères » / « position ». En plus de ces 32 mots, nous en avons choisi huit de plus qui n'apparaissent pas du tout dans la liste. Enfin, nous avons choisi de comparer l'utilisation de ce système de saisie de texte avec le même clavier logiciel sans la liste de mots associée.

Lors de cette étude, il est apparu que l'utilisateur saisit moins rapidement les caractères sur le clavier lorsque ce dernier est couplé à la liste de mots. En effet, quelle que soit la difficulté de la tâche de pointage de la touche, l'utilisateur met en moyenne 100 millisecondes de plus pour saisir le caractère souhaité. Sur la figure 6, les droites rouge et bleue représentent le temps nécessaire pour saisir un caractère, en fonction de la difficulté pour l'atteindre, res-

pectivement avec le clavier azerty et avec le même clavier couplé à une liste de mots. La difficulté de la tâche de pointage d'une touche est déterminée par l'indice de difficulté qui est dépendant de la distance à parcourir pour atteindre la touche et la taille de cette dernière (Cet indice de difficulté sera présenté plus en détail dans la section A du chapitre 4).

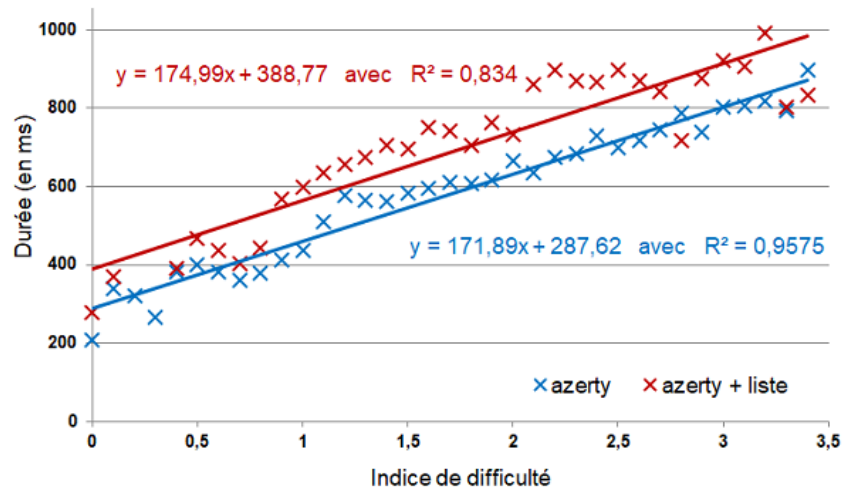


FIGURE 6 – Temps de saisie d'un caractère en fonction de la difficulté pour atteindre la touche correspondante

D'autre part, sur le clavier logiciel avec la liste de prédiction, nous pouvons également constater que le temps moyen pour saisir un caractère augmente proportionnellement au nombre de caractères saisis sur le clavier (cf. Figure 7a). Ce constat s'explique par l'analyse du suivi du regard. Nous pouvons en effet constater que plus l'utilisateur avance dans le mot (et par conséquent saisit des caractères sur le clavier), plus le nombre de fixations du regard sur la liste de mots augmente (cf. Figure 7b).

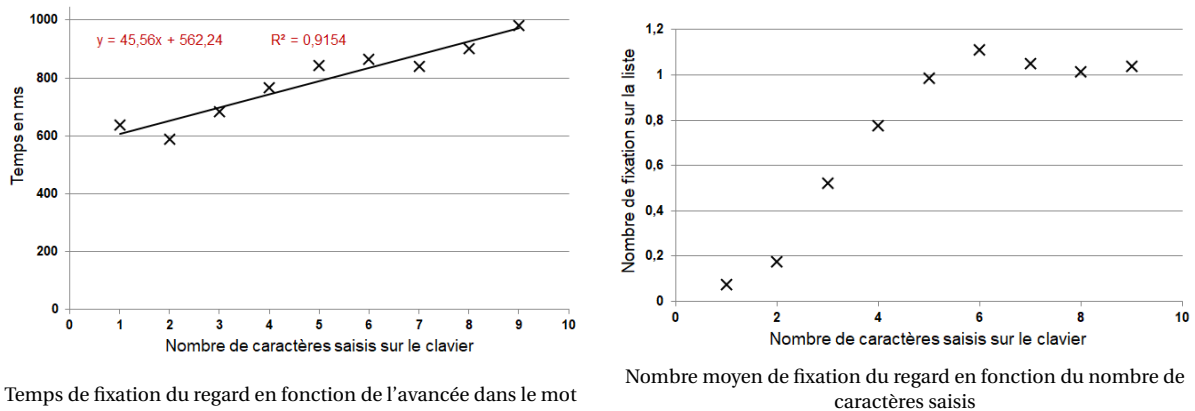


FIGURE 7 – Principaux résultats sur les fixations du regard sur la liste de mots.

On peut également remarquer que, même si le mot apparaît rapidement dans la liste de mots, l'utilisateur sélectionne peu de mots dans la liste avant d'avoir saisi au moins 4 caractères sur le clavier (cf. Figure 8). La liste de mots est surtout utilisée entre le 4e et le 6e caractère saisi.

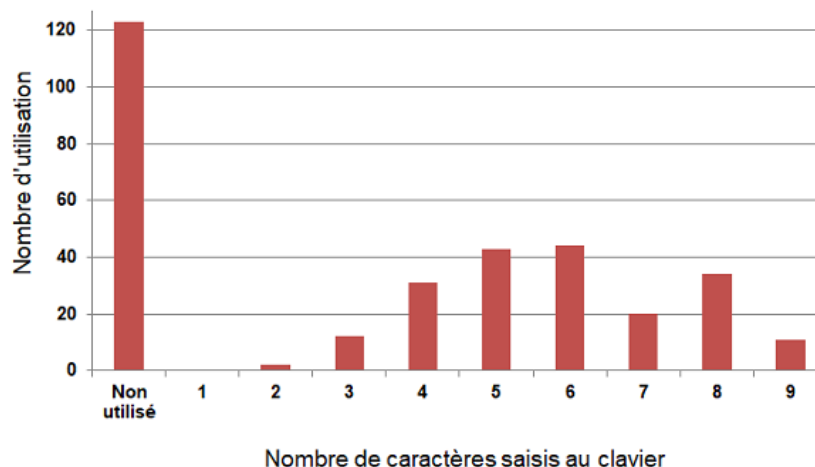


FIGURE 8 – Sélection dans la liste en fonction du nombre de caractères déjà saisis

Enfin, il apparaît également que l'utilisateur a tendance à ne se préoccuper que de la partie supérieure de la liste. D'une part, nous pouvons remarquer que la majorité des fixations effectuées sur la liste se situent sur les premiers éléments de la liste, surtout lorsque l'utilisateur ne fixe qu'une ou deux fois la liste lors de la saisie du mot (cf. Figure 9a). Ceci se traduit également par une sélection des mots dans la liste lorsque ces derniers se trouvent principalement en première ou deuxième position dans la liste (cf. Figure 9b).

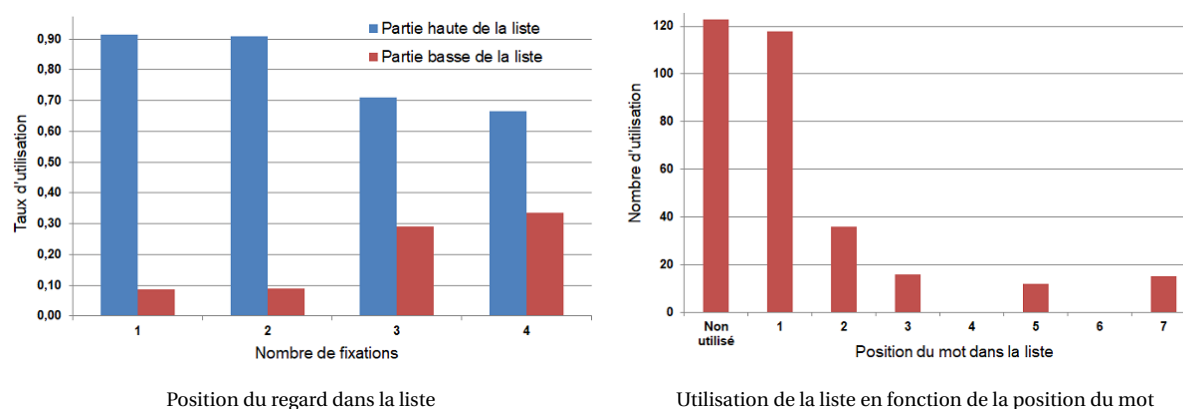


FIGURE 9 – Principaux résultats obtenus lors de l'étude du comportement de l'utilisateur au cours d'une tâche de saisie

Pour résumer les résultats que nous venons de présenter, nous pouvons donc dire qu'en général l'utilisateur ne regarde que le haut de la liste. Par conséquent, il n'utilise pas la liste de

prédiction de manière optimale et saisit plus de caractères sur le clavier logiciel. Cependant, même la saisie sur le clavier logiciel est pénalisée par la présence de la liste à côté et l'attention que lui porte l'utilisateur. Cette utilisation sous optimale de la liste couplée à la saisie moins rapide sur le clavier logiciel entraîne donc une vitesse moyenne de saisie moins rapide sur le dispositif clavier logiciel auquel nous avons ajouté une liste.

B Évaluation théorique des performances sur un clavier avec une liste de mots

Les résultats de l'étude présentée précédemment nous ont amenés à nous pencher sur la manière d'évaluer théoriquement les performances d'un clavier logiciel couplé à une liste de mots.

Les modèles théoriques existants ([W. R. SOUKOREFF et MACKENZIE, 1995; HUGHES et al., 2002] pour les claviers, [SAD et POIRIER, 2009] pour une recherche dans une liste seule, ou [COCKBURN et al., 2007] pour une recherche dans un menu) ne prennent pas en compte dans leur modélisation les constats que nous avons pu formuler lors de notre étude. Le modèle proposé par Soukoreff et Mackenzie [W. R. SOUKOREFF et MACKENZIE, 1995] permet de prédire les performances de saisie sur une disposition de caractères donnée pour un utilisateur expert ou un utilisateur novice. Dans le cas de l'utilisateur expert, le modèle prend simplement en compte l'ensemble des tâches de pointage possible sur le clavier, pondérées par la probabilité qu'elles se produisent. Ceci correspond donc aux performances lors d'une saisie « parfaite » où l'utilisateur serait capable d'enchaîner les pointages de touches sans aucune distraction. A l'autre extrémité, la modélisation des performances d'un utilisateur novice prend en compte qu'après chaque caractère saisi, l'utilisateur cherche sur le clavier le nouveau caractère qu'il souhaite saisir. Ce temps de recherche visuelle (modélisé par la loi de Hick-Hyman [HICK, 1952; HYMAN, 1953]) vient s'ajouter au temps de pointage calculé de la même manière que pour les utilisateurs experts. Le modèle de Soukoreff et Mackenzie propose donc les bornes possibles des performances que peut espérer avoir un utilisateur avec une disposition de touches donnée.

Dans notre cas, nous envisageons une valeur intermédiaire car l'utilisateur ne découvre pas la disposition, donc n'est pas novice. Même s'il connaît très bien la disposition des touches, il ne pourra pas atteindre les performances d'un expert car ses tâches de pointage seront perturbées par l'attention qu'il portera sur la liste de mots. L'objectif est donc de modéliser le temps que prend l'attention portée sur la liste des mots. L'autre challenge est que, à la différence des dispositions de caractères évaluées avec le modèle de Soukoreff et Mackenzie qui sont statiques, dans notre cas, la liste de mots est dynamique.

Le modèle que nous proposons a pour objectif de prédire le temps moyen nécessaire pour saisir un caractère lorsque nous effectuons une saisie au moyen d'un clavier logiciel couplé à une liste de prédiction. La saisie des mots peut être dépendante de la liste de mots présents. Notre modèle est forcément dépendant du système de prédiction proposé. Pour calculer le temps moyen de saisie d'un caractère, nous nous appuyons sur le temps mis en moyenne pour saisir un ensemble de mots d'un corpus. Pour chaque mot, nous calculons le temps moyen nécessaire pour saisir ce mot. Le temps moyen pour saisir un caractère avec le système évalué est la somme des temps prédits pour chaque mot divisé par le nombre total de caractères composant l'ensemble des mots du corpus :

$$T = \frac{\sum_{m \in D} T(m)}{\sum_{m \in D} |m|} \quad (2.1)$$

Avec D qui représente le dictionnaire de mots considéré, alors que $|m|$ détermine la taille du mot m (en nombre de caractères), et $T(m)$ est le temps moyen mis pour saisir le mot m .

Le temps de saisie d'un mot est dépendant de la présence ou non du mot souhaité dans la liste de mots proposés, et si tel est le cas, à partir de combien de caractères saisis au clavier le mot apparaît dans la liste. Pour prédire ce temps de saisie, nous avons choisi de prendre en compte tous les cas de saisie possible. Par exemple, pour un mot de 8 caractères, si le mot apparaît dans la liste après que l'utilisateur ait saisi 4 caractères, nous allons prendre en compte le cas où l'utilisateur sélectionne le mot dans la liste dès qu'il apparaît, mais aussi les cas où il saisit respectivement un, deux ou trois caractères supplémentaires sur le clavier avant de sélectionner le mot dans la liste. Enfin, nous prenons également en compte le cas où il saisit tout le mot sur le clavier. Ainsi, le temps moyen pour saisir un mot est obtenu au moyen de la formule suivante :

$$T(m) = \frac{1}{1 + |m| - P(m)} \sum_{i=P(m)} T(m)_i \quad (2.2)$$

Où $P(m)$ correspond au nombre de caractères à partir duquel le mot apparaît dans la liste, et $T(m)_i$ est le temps nécessaire pour saisir le mot m en considérant que l'utilisateur utilisera la liste de prédiction après le i^{eme} caractère saisi au clavier. Dans ce cas, le temps de saisie est obtenu en sommant l'ensemble des tâches de pointage sur le clavier logiciel ainsi que le temps nécessaire pour sélectionner l'item dans la liste s'il est choisi. On obtient la formule suivante :

$$T(m)_n = \sum_{i=1} (T(m)_i + E(i, |m|) \times TL) \quad (2.3)$$

Où, TL est une constante représentant le temps de sélection d'un item dans la liste de prédiction. Ce temps a été fixé à 1,26 s et correspond au temps moyen de sélection dans la liste que nous avons obtenu dans l'expérimentation présentée à la section précédente. Enfin, $T_i(m)$ correspond au temps nécessaire pour aller pointer sur le clavier le i^{eme} caractère du mot m sachant que l'on vient du $(i-1)^{eme}$ caractère. Ce temps est obtenu au moyen de la loi de Fitts :

$$T_i = a + b \times \log_2 \left(\frac{D_i(m)}{W} + 1 \right) \quad (2.4)$$

Où $D_i(m)$ est la distance qui sépare les $(i-1)^{eme}$ et i^{eme} caractères du mot m . Les constantes a et b ont respectivement été fixées à 0,390 et 0,175s conformément à la droite de régression linéaire présentée sur la Figure 6. Enfin, pour définir si le mot apparaît dans la liste des mots prédits, et peut donc être sélectionné, nous utilisons :

$$E(i, |m|) = \begin{cases} 1 & \text{si } i < |m| \\ 0 & \text{sinon.} \end{cases} \quad (2.5)$$

Les claviers logiciels augmentés d'une liste de prédiction sont dépendants du système de prédiction qu'ils utilisent pour afficher les résultats de prédiction. Par conséquent, pour pouvoir évaluer théoriquement un clavier de ce type, il nous faut connaître le système de prédiction utilisé. De plus, à la différence du modèle de Soukoreff et Mackenzie, la fonction $P(m)$ est plus qu'une simple table de fréquences de bigramme. En effet, il paraît assez difficile d'avoir une table qui représente l'ensemble des fréquences d'apparition des mots en fonction du préfixe saisi.

De ce fait, notre modèle ne peut pas être vu comme une équation mathématique mais doit plus être considéré comme un modèle décrivant l'algorithme à mettre en place pour connaître le temps moyen de saisie. Cet algorithme, pour pouvoir fonctionner, a besoin d'interagir avec le système de prédiction de mots qui serait utilisé par le clavier logiciel. Il a également besoin de connaître la position des différentes touches et items de la liste, ainsi que la disposition des caractères sur les différentes touches. Cet algorithme simule alors la saisie de l'ensemble des mots du dictionnaire proposé en explorant l'ensemble des possibilités qui sont offertes à l'utilisateur.

Afin de pouvoir paramétrer facilement notre algorithme, nous avons conçu une interface permettant d'éditer la configuration des touches et de la liste, choisir le dictionnaire de mots, ainsi que l'algorithme de prédiction à utiliser (cf. Figure 10).

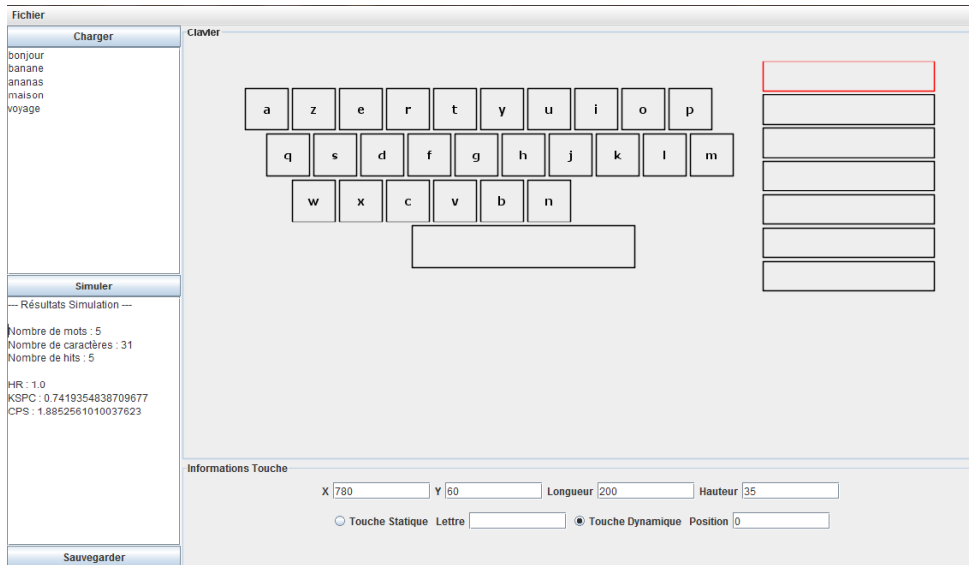


FIGURE 10 – Interface de prototypage et d'évaluation de clavier logiciel et liste de prédiction

Une fois le clavier prototypé sur l'interface, nous pouvons effectuer une simulation dite « parfaite » de l'ensemble des mots choisis. La simulation d'une saisie parfaite est une technique évaluant les performances théoriques d'un système d'entrée de texte. On l'appelle « parfaite » parce qu'elle ne prend pas en considération la fatigue et les erreurs commises par l'utilisateur durant la saisie. Dans notre cas, cette évaluation calcule les performances de saisie au moyen des formules présentées précédemment.

C Synthèse et perspectives

J'ai présenté, dans ce chapitre, mes travaux théoriques en lien avec l'évaluation des claviers logiciels. Ils se sont principalement déroulés en deux temps : dans un premier temps, une étude sur l'utilisation d'une liste de mots lors de la saisie de texte avec un clavier logiciel. Cette étude a mis en évidence que les utilisateurs peuvent être perturbés par l'utilisation d'une liste de mots. En effet, le temps pour saisir un caractère sur le clavier est augmenté d'en moyenne 100 millisecondes lorsqu'une liste de mots est présente à côté du clavier. D'autre part, l'analyse du suivi du regard, nous a également permis de constater que les utilisateurs ne prenaient principalement connaissance que des mots qui se trouvent en haut de la liste. Ces travaux ont été l'objet d'une partie de la thèse de Georges BADR, doctorant que j'ai encadré de 2007 à 2011.

Les résultats de cette étude, détaillés dans le mémoire de thèse de Georges [BADR, 2011], nous ont permis de proposer, dans un second temps, un modèle de prédiction de la vitesse de saisie de texte sur un clavier logiciel couplée à une liste de prédiction. Pour aider les concep-

teurs à utiliser ce modèle, nous proposons en complément un outil de simulation qui utilise ce modèle pour prédire les performances d'un prototype de clavier logiciel.

Le modèle que nous proposons est une première base pour la prédiction de la vitesse de saisie sur ce type de clavier logiciel. Elle prend en compte le fait que l'utilisateur a plusieurs alternatives possibles pour saisir un même mot. Cependant, du fait du faible nombre de données à notre disposition, nous avons choisi de fixer deux paramètres de notre modèle :

1. La probabilité que l'utilisateur choisisse une alternative plutôt qu'une autre. Actuellement, dans notre modèle, nous prenons en compte chaque alternative de manière équiprobable. Les résultats de l'étude sur le suivi du regard au cours de la saisie montrent que l'utilisateur ne va pas forcément prendre l'alternative la plus avantageuse, mais continuera de saisir sur le clavier jusqu'à ce que le mot souhaité apparaisse dans les premières positions de la liste. Par conséquent, les différentes alternatives ne sont certainement pas utilisées de manière équiprobable. Il serait donc intéressant d'analyser la probabilité d'utilisation d'un mot en fonction de sa position dans la liste. Cependant, nos données d'expérimentation ne suffisent pas pour proposer un modèle probabiliste qui prendrait en compte la position du mot dans la liste, et le nombre de caractères saisis avant l'apparition du mot dans la liste. Il serait donc nécessaire de réaliser de nouvelles évaluation nous permettant de collecter beaucoup plus de données sur le suivi du regard.
2. Nous utilisons une constante pour déterminer le temps de sélection d'un item dans la liste. Cette sélection est principalement un pointage effectué par l'utilisateur après avoir saisi un ensemble de caractères sur le clavier. Ce temps de pointage peut être prédit en utilisant la loi de Fitts. Mais pour effectuer une prédiction précise, il faut avoir des données de pointage pour un intervalle d'indices de difficulté assez large. Le nombre de données de pointage sur la liste de mots ne nous permettait pas d'avoir suffisamment d'information pour proposer un modèle théorique de pointage assez précis.

Ainsi, pour pouvoir affiner notre modèle théorique, il serait nécessaire de faire à nouveau des sessions d'évaluation de clavier logiciel couplé à une liste de prédiction pour recueillir un ensemble de données plus conséquent.

Chapitre 3

Intégration des résultats de la prédiction pour une meilleure interaction

Comme nous pouvons le constater dans les différents états de l'art sur les systèmes de saisie de texte [DUBE et ARIF, 2019; LUNA et al., 2018; MACKENZIE et R. W. SOUKOREFF, 2002; MARTIN et PECCI, 2007; POIRIER et al., 2018; SIQUEIRA et al., 2016], la majorité des claviers logiciels intègrent des algorithmes utilisant des connaissances linguistiques pour optimiser la saisie. Il existe principalement 3 types d'algorithmes : 1) Les systèmes de prédiction; 2) Les systèmes de correction; 3) Les systèmes de déduction.

Une partie de mes travaux ont porté sur, d'une part, l'optimisation de l'intégration des résultats de l'algorithme de prédiction de mots et de caractères dans les claviers logiciels. Ces travaux avaient pour objectif d'améliorer les performances de saisie des personnes ayant une déficience motrice. D'autre part, je me suis intéressé au système de déduction pour faciliter la saisie de texte sur écran tactile pour les utilisateurs déficients visuels.

Dans ce chapitre, je vous présente les différents travaux que nous avons réalisés pour répondre à ces axes de recherche.

A Intégration de la prédiction de mots

Comme nous avons pu le voir dans le chapitre précédent, le principal problème avec l'utilisation des listes de prédiction est le temps que l'utilisateur perd à lire la liste des mots proposés alors que celle-ci ne contient pas le mot qu'il souhaite saisir. La problématique est donc de limiter ces moments où l'utilisateur regarde la liste sans pour autant s'en servir.

Pour augmenter le nombre de fois où l'utilisateur va interagir avec la liste de mots, nous proposons le système WordTree. WordTree est un composant graphique présentant une liste de

mots qui peut être couplée à tout clavier logiciel. Comme pour toute liste de mots prédits, WordTree utilise un algorithme de prédiction de mots qui, à partir du préfixe saisi propose les mots les plus probables. L'originalité de WordTree est que l'utilisateur n'est pas obligé de sélectionner un mot complet de la liste. Il a la possibilité de sélectionner qu'une partie du mot. Pour cela, chaque caractère des mots proposés dans la liste est associé à une zone interactive et est donc sélectionnable. Lorsque l'utilisateur sélectionne un caractère dans le mot, toute la chaîne de caractères comprise entre le préfixe déjà saisi et le caractère sélectionné est insérée. Par exemple, sur la Figure 11, l'utilisateur souhaite saisir le mot « bouillir », et a déjà saisi le préfixe « bo ». Le mot souhaité n'est pas dans la liste, mais en sélectionnant le second 'L' de « bouillon », l'utilisateur saisit en une action la chaîne de caractère « uill ». Notre hypothèse était que permettre de ne sélectionner ne serait-ce qu'une partie du mot permettrait de diminuer le nombre d'actions à effectuer par l'utilisateur. D'autre part, en sélectionnant une partie du mot, l'utilisateur ne perd pas complètement le temps qu'il a passé à lire la liste des mots. Enfin, plus on avance dans le mot à saisir, et plus il y a de chance que l'utilisateur puisse compléter son mot avec les mots (ou parties de mots) proposés dans WordTree. L'utilisateur va donc porter son attention sur WordTree et aura moins d' « allers / retours » à effectuer entre le clavier et WordTree.



FIGURE 11 – WordTree après la saisie du préfixe "bo"

Une évaluation par simulation de la saisie de texte nous a permis d'estimer les performances théoriques du système WordTree par rapport à une liste de prédiction classique. Cette évaluation consistait à simuler la saisie « parfaite » des 364 370 mots de l'Officiel du Scrabble. Une saisie parfaite est une saisie où l'ordinateur va privilégier l'utilisation de la liste de mots (ici, WordTree) plutôt que la saisie des caractères au moyen du clavier logiciel. Les résultats de la simulation sont synthétisés dans le Tableau 1. Ces résultats théoriques montrent que le système WordTree permettrait d'augmenter le nombre de fois où l'utilisateur pourra utiliser les résultats de l'algorithme de prédiction. Pour cela, nous mesurons le Hit Rate (HR) [FAZLY et HIRST, 2003], qui est le pourcentage d'apparition du mot à saisir dans la liste. Dans

le cas de WordTree, nous calculons le HR comme le nombre de fois où l'utilisateur va pouvoir compléter le préfixe déjà saisi en sélectionnant, au moins, une partie du mot dans la liste.

Ces premiers résultats ont ensuite été confirmés par deux évaluations avec des utilisateurs. Dans un premier temps, nous avons réalisé une évaluation comparative avec 16 sujets sains, puis dans un second temps, le même type d'évaluation a été réalisé avec 9 participants ayant une déficience des membres supérieurs. Le type de tâche était le même pour les 2 évaluations : recopier une série de mots proposés aux participants. D'après les résultats, résumés dans le Tableau 1, on peut voir que le nombre de frappes effectuées en moyenne pour saisir un caractère (en anglais, « KeyStrokes Per Character » (KSPC) [MACKENZIE, 2002]) a bien diminué dans les deux cas, et se traduit par une augmentation significative de la vitesse de saisie de texte (calculée en nombre de caractères saisis par seconde (CPS)). Le gain de vitesse est d'environ 20,6 % pour les sujets sains ($F_{1,30} = 4,18$ avec $p = 0,049$), et de 21,4 % pour les sujets déficients des membres supérieurs ($F_{1,16} = 7,4207$ et $p = 0,015$). Les résultats plus détaillés de ces 2 études sont disponibles dans [BADR et RAYNAL, 2009c].

TABLEAU 1 – Résultats des évaluations comparant le système WordTree et la liste de prédiction de mots

	Simulation		Sujets sains		Sujets déficients moteur	
	KSPC	HR	CPS	KSPC	CPS	KSPC
Liste de prédiction classique	0,92	3,7 %	0,92	0,90	0,42	0,47
WordTree	0,71	18,7 %	1,11	0,75	0,51	0,65

Ces travaux avec le système WordTree ont été complétés par la réalisation d'un second prototype. En effet, un des problèmes relevés par notre étude sur l'utilisation des listes de prédiction est le fait que l'utilisateur doit détourner son attention du clavier pour regarder la liste des mots proposés sur le côté du clavier. Pour pallier ce problème, nous avons proposé le clavier Centralist [BADR et RAYNAL, 2011] qui dispose la liste de mots prédits au centre du clavier. Les touches sont disposées tout autour de la liste. La disposition des touches étant assez éloignée de la configuration classique d'un clavier type AZERTY ou QWERTY où les caractères sont disposés sur 3 lignes, nous avons fait le choix de positionner les caractères par ordre alphabétique pour faciliter la recherche de ces derniers. L'espace étant le caractère le plus utilisé, nous avons positionné une barre espace sur chaque côté du clavier (cf. Figure 12). Enfin, la liste de mots proposée au centre du clavier peut être de tout type. Nous avons

choisi d'utiliser ici le système WordTree pour inciter l'utilisateur à utiliser cette liste le plus souvent possible.

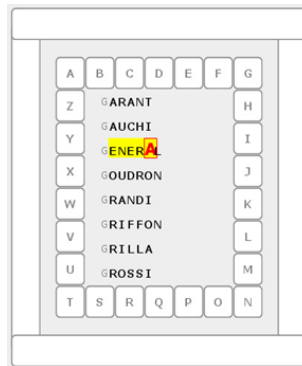


FIGURE 12 – Disposition du clavier CentraList

Pour évaluer ce nouveau prototype, nous avons également effectué deux évaluations : une première avec 18 participants sans déficience motrice, puis une seconde avec 7 participants ayant une déficience motrice des membres supérieurs. Les participants avaient deux exercices à réaliser : un avec CentraList et un second avec un clavier logiciel avec une disposition des caractères par ordre alphabétique accompagné du système WordTree disposé à sa droite. Les participants avaient pour consigne de saisir le plus rapidement possible de courtes phrases (25 phrases pour les participants sans déficience et 15 pour ceux avec une déficience).

Cette évaluation comparative nous a permis de montrer que, du fait de sa disposition, CentraList nécessitait moins de déplacement du pointeur pour saisir le texte (colonne Dist dans le Tableau 2). En effet, la distance moyenne entre 2 opérations (sélection d'un caractère sur le clavier ou d'un mot dans la liste) est réduite de près de 10% pour les sujets sains et d'environ 23 % pour les participants ayant une déficience motrice. La position de la liste au centre du regard de l'utilisateur a comme effet d'augmenter l'utilisation de la liste par les utilisateurs. Les participants ont plus sélectionné de mots de la liste avec CentraList qu'avec l'autre clavier.

D'autre part, l'utilisation plus importante de la liste permet de réduire le nombre d'actions nécessaires pour saisir un caractère (KSPC). Les participants ont réalisé moins d'actions avec CentraList (0,82 pour les personnes valides et 0,77 pour les personnes avec un handicap moteur) qu'avec le clavier Alphabétique classique et la liste sur le côté (0,84 pour les participants valides et 0,8 pour les autres participants). La liste de prédiction et le système de prédiction étant les mêmes pour les deux claviers, cela veut donc dire que les participants ont eu plus souvent recours à la liste avec CentraList que sur le clavier alphabétique avec la liste sur le côté.

Ces deux facteurs (diminution des distances et augmentation de l'utilisation de la liste) ont eu un effet bénéfique sur la vitesse de saisie de texte. Les participants valides ont saisi en moyenne 0,90 caractères par seconde avec CentraList contre 0,81 avec le clavier alphabétique, soit une augmentation de 11,1 %. Pour les participants en situation d'handicap, la vitesse est passée de 0,46 caractère par seconde à 0,52 avec CentraList soit une hausse de 13 %. Ce clavier présente l'avantage d'attirer l'attention de l'utilisateur sur la liste. Cette attention plus grande permet ainsi à l'utilisateur de voir plus fréquemment les mots ou les chaînes de caractères de la liste qui peuvent l'aider dans sa saisie. Ainsi, il utilise plus souvent la liste, ce qui a pour avantage de diminuer le nombre d'actions à réaliser. Ceci, couplé au fait que les distances à parcourir avec le pointeur sont moins longues, entraîne une augmentation de la vitesse de saisie de texte.

Ces deux derniers facteurs sont notamment très importants pour les personnes ayant une faible motricité des membres supérieurs car la diminution de la distance à parcourir leur permet de diminuer la fatigue motrice. D'autre part, une augmentation de la vitesse de saisie est toujours bienvenue pour ces personnes dont la vitesse de saisie est généralement largement inférieure à celle que des personnes à la motricité normale peuvent avoir sur un clavier physique.

TABLEAU 2 – Résultats des évaluations comparant le clavier CentraList et le clavier alpha

	Sujets sains				Sujets déficients moteur			
	Dist	HR	CPS	KSPC	Dist	HR	CPS	KSPC
Clavier Alpha	336.78	15,88	0.81	0.84	460.54	19.8	0.46	0.8
CentraList	304.4	17,4	0.9	0.82	353.51	22.2	0.52	0.77

B Intégration de la prédiction de caractères

Les algorithmes de prédiction de caractères sont moins répandus que ceux basés sur la prédiction de mots. Ils sont surtout utilisés dans le cas des claviers logiciels utilisant le défilement automatique du curseur. Dans ce cas précis, le curseur se déplace de touche en touche, généralement de gauche à droite et de bas en haut sur le clavier, à intervalle régulier. Après chaque caractère saisi, le curseur est toujours repositionné dans le coin supérieur gauche du clavier. L'utilisateur doit alors attendre que le curseur arrive sur le caractère souhaité pour le valider au moyen d'un contacteur. L'algorithme de prédiction de caractères est donc utilisé pour ré-agencer les caractères sur le clavier en fonction de leur probabilité d'être le sui-

vant. Ainsi, les caractères les plus probables sont positionnés au plus près du coin supérieur gauche pour limiter le nombre de déplacements que devra effectuer le curseur.

Dans le cas des claviers logiciels où l'on utilise un dispositif de pointage pour manipuler un pointeur, la fréquence d'apparition des caractères et des bigrammes est surtout utilisée pour optimiser la disposition des caractères sur le clavier (par exemple, [RAYNAL et VIGOUROUX, 2005a; ZHAI et al., 2000] pour une optimisation de la disposition des caractères pour l'anglais et [RAYNAL, 2006] pour le français). Mais le ré-agencement dynamique des caractères, en cours de saisie, n'est pas envisageable car à la différence du curseur, le pointeur n'est pas repositionné après chaque caractère. Un ré-agencement des caractères imposerait un temps de recherche visuel trop important entre chaque caractère à saisir pour l'utilisateur. L'utilisateur ne pourrait alors plus anticiper ses déplacements d'un caractère à un autre.

Les algorithmes de prédiction de caractères ont toutefois été utilisés pour ce type de clavier. Magnien et al. [MAGNIEN et al., 2004] a notamment utilisé la prédiction de caractères pour mettre en évidence les caractères qui ont le plus de chance d'être saisis. L'objectif étant d'attirer l'attention d'un utilisateur novice avec la disposition des caractères vers ces caractères qui ont le plus de chance d'être saisis.

Mais l'autre axe d'amélioration possible au moyen d'un algorithme de prédiction de caractères est l'optimisation de la phase de pointage. D'après la loi de Fitts [FITTS, 1954], le temps nécessaire pour pointer une cible est dépendant de la distance pour atteindre cette cible et de la taille de cette dernière. Cette loi est notamment utilisée pour prédire les performances théoriques que pourrait avoir un utilisateur avec un clavier donné [W. R. SOUKOREFF et MACKENZIE, 1995], et a été utilisée à de nombreuses reprises pour proposer des claviers avec des dispositions de caractères optimisées pour une saisie au doigt ou avec un pointeur [RAYNAL, 2006].

Il est donc possible d'améliorer la tâche de pointage de 2 manières : soit en réduisant la distance à parcourir, soit en augmentant la taille de la cible. Ces optimisations se basent à chaque fois sur l'utilisation d'un algorithme de prédiction de caractères qui donne pour chaque caractère la probabilité qu'il soit le prochain saisi. L'objectif est ainsi de faciliter le pointage vers les caractères les plus probables.

Le système KeyGlass que j'ai conçu durant ma thèse [RAYNAL, 2004; RAYNAL et VIGOUROUX, 2005b] a pour but de diminuer la distance à parcourir entre chaque caractère à saisir. Pour cela, après chaque caractère saisi par l'utilisateur, 4 touches supplémentaires apparaissent autour de la dernière touche sélectionnée. Un algorithme de prédiction de caractères est utilisé pour associer à ces 4 nouvelles touches les caractères qui ont le plus de chance de succéder au préfixe déjà saisi (cf. Figure 13). La simulation d'une saisie parfaite prévoyait un gain théorique de pratiquement 60% sur la distance parcourue. Par conséquent, d'après la loi de

Fitts, la vitesse de saisie de texte pour un utilisateur expert peut théoriquement augmenter d'environ 43% si l'utilisateur utilise les 4 touches supplémentaires à chaque fois que celles-ci sont proposées.

Cependant, nous avons pu constater lors des évaluations avec les utilisateurs que le gain de temps escompté n'était pas présent [RAYNAL, 2007]. En effet, ces évaluations ont mis en évidence que les utilisateurs utilisaient bien ces touches supplémentaires (dans 96% des cas où le caractère est proposé sur une des touches supplémentaires), et leur utilisation a bien permis de réduire les distances de 53,8%. En revanche, cette diminution des distances parcourues n'a pas suffi pour augmenter la vitesse de saisie de texte (augmentation du temps pour saisir un mot de 20,1%). Notre hypothèse est que, de la même manière qu'avec la liste de mots, l'ajout des touches supplémentaires perturbe l'utilisateur. En effet, nous avons pu constater que les utilisateurs anticipent la saisie du caractère suivant. Le fait de leur proposer des caractères supplémentaires les obligent à prendre connaissance de ces nouveaux caractères, et par conséquent les coupent dans leur dynamique, leur anticipation impactant alors négativement leur performance.

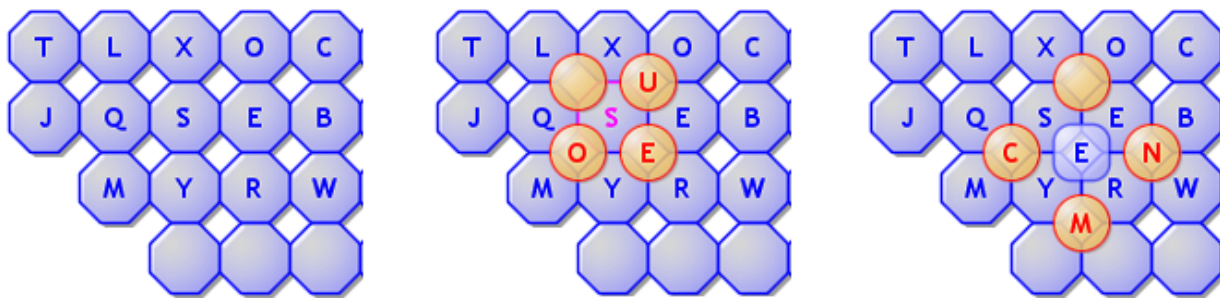


FIGURE 13 – Clavier pour la version 2 des KeyGlasses : A. clavier simple ; B. KeyGlasses qui apparaissent après la saisie du 's' ; C. KeyGlasses qui apparaissent après la saisie du 'e' qui était placé sur une KeyGlass. Extrait de [RAYNAL, 2014]

Ce problème n'est pas présent avec un clavier à défilement automatique car l'utilisateur est dépendant du défilement du curseur et ne peut donc pas anticiper la saisie des caractères. Ce temps de défilement peut également lui permettre de regarder la nouvelle disposition sans que cela n'ait d'incidence sur le temps de saisie.

La problématique quant à l'intégration des résultats de la prédiction de caractères est donc d'intégrer ces résultats dans le clavier sans que cela ne vienne perturber l'utilisateur dans sa tâche de saisie. L'ajout de touches pour diminuer la distance à parcourir ayant montré ses limites, mes travaux ont porté sur l'étude des autres alternatives permettant de faciliter la tâche de pointage sans modifier la position initiale des touches et des caractères sur le clavier. J'ai donc, dans un premier temps, exploré les solutions consistant à modifier la taille des touches.

Quelques travaux ont déjà été menés sur cet axe de recherche. Le plus connu d'entre eux est le clavier Dasher [WARD et al., 2000], sur lequel tous les caractères sont affichés verticalement sur la droite de l'écran, et défilent de droite à gauche. Pour sélectionner un caractère, l'utilisateur déplace un pointeur sur l'axe vertical au centre de l'écran. Au fur et à mesure de la sélection des caractères, les caractères apparaissent à nouveau sur la droite du clavier. L'algorithme de prédiction de caractères permet ici de moduler la taille des touches en fonction de la probabilité de saisir le caractère lié à celle-ci. Les caractères les plus probables sont donc plus facilement sélectionnables du fait de la taille de la zone sur laquelle ils se trouvent. Cependant ce système de saisie de texte est peu utilisé car il nécessite une attention importante tout au long de la saisie avec ce dispositif.

Le clavier BigKey [AL FARAJ et al., 2009] joue lui sur la taille des touches, sans toutefois modifier leur position et la disposition des caractères sur celles-ci. A la différence des claviers Fisheye [RAYNAL et TRUILLET, 2006; RAYNAL et al., 2007; RAYNAL et TRUILLET, 2007] et HoverZoom [POLLMANN et al., 2014], dont la taille des touches dépend de la position du pointeur, le clavier BigKey modifie la taille des touches des caractères les plus probables. Ainsi, après chaque caractère saisi, les touches liées aux 4 caractères les plus probables sont agrandies. L'inconvénient du clavier BigKey est que les touches agrandies recouvrent en partie les touches voisines. Ceci peut engendrer une gêne lors de la sélection de ces touches.

Pour remédier à ce problème, une collaboration avec Benoit Martin de l'université de Metz nous a amenés à proposer le clavier FloodKey. Ce clavier modifie la forme des touches et la surface de chacune d'elle en fonction de la probabilité de chaque caractère (cf. Figure 14). Pour réaliser cette disposition de touches sans chevauchement, nous utilisons un diagramme de Voronoi couplé à l'algorithme de Fruchterman-Reingold [FRUCHTERMAN et REINGOLD, 1991]. Cependant, les modifications de la forme des touches et de leur surface d'affichage tout au long de la saisie sont apparues trop perturbantes pour les utilisateurs. Les premiers résultats n'ont pas été concluants [AULAGNER et al., 2010].

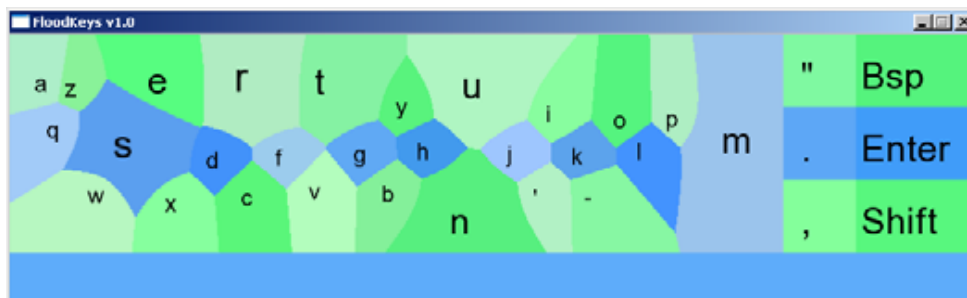


FIGURE 14 – Disposition des touches sur Floodkey après la saisie de « joue »

En parallèle de FloodKey, nous avons également testé une autre manière d'élargir la surface de sélection des caractères les plus probables. Durant ses travaux de thèse, Bruno Merlin a proposé le système SpreadKey [MERLIN et RAYNAL, 2009]. Le principe est de « propager » un

caractère sur les touches voisines à sa touche d'origine lorsque sa probabilité d'être saisi est bien plus importante que celle des caractères des touches voisines (cf. Figure 15). Dans le cas où un caractère est propagé sur les touches voisines, celui-ci peut être saisi en pointant et cliquant sur sa touche d'origine, mais également sur les touches sur lesquelles il s'est propagé. Cependant, chaque caractère reste présent et accessible sur sa touche d'origine quelle que soit sa probabilité d'être saisi. Dans le cas où un caractère voisin se serait propagé sur sa touche, le caractère initial est affiché en second plan dans le coin supérieur gauche de la touche et est sélectionnable via un long tap ou un geste en direction du coin supérieur gauche de la touche.

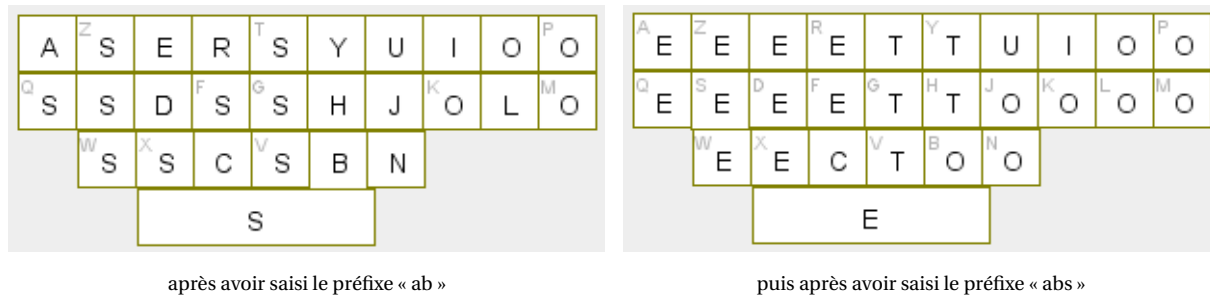


FIGURE 15 – Disposition des caractères sur le clavier Spreadkey

Pour ce clavier, nous avons réalisé une expérimentation avec 6 participants déficients des membres supérieurs. Tous utilisaient un trackball comme dispositif de pointage. Pour que chaque participant ait le temps de s'appropriier complètement le système SpreadKey, chacun a réalisé 10 sessions, à raison de 2 sessions par jour, espacées de 3 à 5 heures, pendant 5 jours. Chaque session consistait à recopier un ensemble de phrases courtes avec le clavier logiciel QWERTY¹, avec et sans le système SpreadKey.

Les principaux résultats que nous avons obtenus sont : d'une part une réduction de la distance parcourue par le pointeur : la distance moyenne parcourue pour saisir un mot sur le clavier sans SpreadKey était de 213 pixels contre 138 pixels avec le système SpreadKey, soit une réduction de la distance d'environ 35,2 %. D'autre part, les utilisateurs ont également saisi plus rapidement sur le clavier augmenté du système SpreadKey : leur vitesse de saisie était respectivement de 13,08 mots par minute et 10,44 mots par minute avec et sans le système SpreadKey, soit une augmentation de 25,3 %. De plus, la Figure 16 présente l'évolution de la vitesse de saisie au cours des 10 sessions pour le clavier seul et accompagné du système SpreadKey. Sur cette figure, nous pouvons constater que dès la première session les utilisateurs saisissaient plus rapidement avec SpreadKey qu'avec le clavier seul. L'écart de vitesse entre les 2 systèmes a eu tendance à augmenter au cours des sessions. Enfin, l'analyse des

1. L'expérimentation s'est déroulée au Brésil. La disposition standard au Brésil est la disposition QWERTY. L'ensemble des phrases à saisir étaient en portugais et avaient été choisies sur le même principe que celles construites pour le corpus de phrase de Mackenzie et Soukoreff [MACKENZIE et R. W. SOUKOREFF, 2003]

erreurs de saisie nous a permis d'identifier que les utilisateurs effectuaient un peu plus d'erreurs avec le système SpreadKey qu'avec le clavier seul (le taux d'erreur est de 1,5 % avec le clavier seul contre 2,6 % avec SpreadKey). Cependant le taux d'erreurs reste inférieur au taux d'erreur acceptable de 3 %. Tous ces résultats ont été publiés à la conférence ICCHP 2010 [MERLIN et RAYNAL, 2010].

En résumé, dans le cas de SpreadKey, les résultats de l'algorithme de prédiction de caractères ont été intégrés au clavier sans que cela ne perturbe de manière conséquente l'utilisateur. Ceci est dû au fait que les touches n'étaient pas modifiées. La disposition des caractères sur celles-ci évoluait au cours de la saisie, mais le caractère restait toujours au minimum sur sa touche d'origine. Par conséquent, même si la disposition des caractères changeait, l'utilisateur pouvait quand même effectuer le mouvement qu'il avait anticipé, et réajuster la fin du déplacement du pointeur si le caractère souhaité était propagé sur les touches voisines.

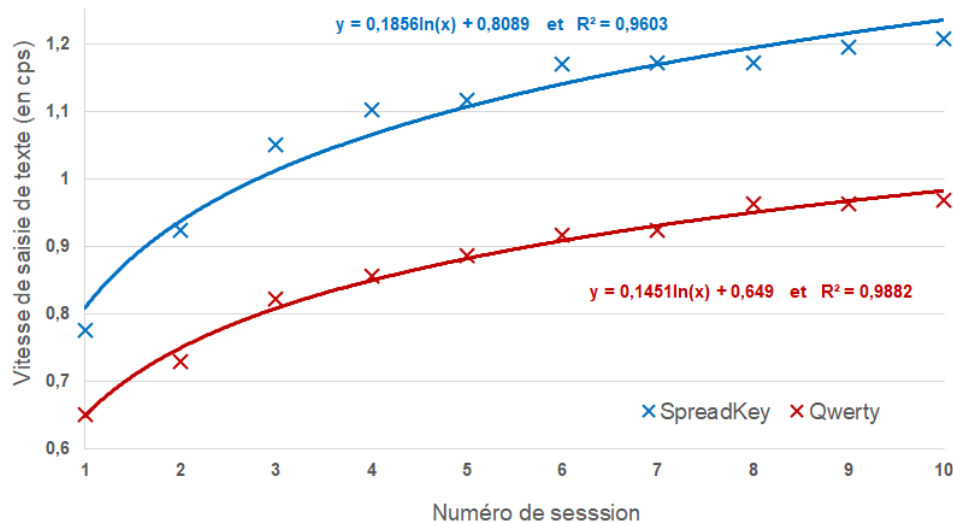


FIGURE 16 – Vitesse de saisie de texte (en caractères par seconde) au cours des 10 premières sessions avec le simple clavier QWERTY et augmenté du système SpreadKey

Enfin, pour finir d'explorer les différentes alternatives possibles pour améliorer la phase de pointage grâce aux résultats de l'algorithme de prédiction de caractères, je me suis intéressé à la modification de la fonction de transfert. Le principe est celui du pointage sémantique proposé par Renaud Blanch et al. [BLANCH et al., 2004]. Il consiste à moduler le rapport entre le déplacement à l'écran et celui dans l'espace physique en fonction de ce qui se trouve sous le pointeur. S'il n'y a pas d'éléments pertinents sous le pointeur, le déplacement du pointeur est accéléré, et inversement, si le pointeur se trouve sur un objet d'intérêt, sa vitesse de déplacement sera réduite. L'objectif est de faciliter le déplacement d'une cible à une autre et d'améliorer la précision de pointage d'un objet. Nous avons repris ce principe pour le clavier

sémantique en donnant un poids à chaque touche en fonction de la probabilité de leur caractère d'être saisi. La vitesse de déplacement du pointeur sur le clavier est alors liée au poids de chaque touche : plus la touche a un faible poids, plus le pointeur passe rapidement dessus, et plus la touche a un poids élevé, plus le pointeur se déplace lentement sur la touche. L'objectif est ainsi de passer rapidement sur les touches qui ont peu de chance d'être sélectionnées et de faciliter le pointage sur celles dont le caractère a une forte probabilité d'être saisi. Le pointage sémantique a l'avantage de ne pas modifier l'apparence du clavier. L'aspect graphique du clavier reste le même tout au long de la saisie, seul le ratio de déplacement entre le dispositif de pointage et le pointeur est modifié. Ainsi, par rapport aux solutions explorées jusqu'à présent, l'utilisateur ne sera pas perturbé par les changements dynamiques effectués sur le clavier.

Pour évaluer la pertinence de cette approche, nous avons réalisé une première expérimentation avec 12 participants. Il leur était demandé de réaliser 2 exercices de saisie avec d'une part un clavier avec la disposition traditionnelle AZERTY, et d'autre part, le même clavier augmenté du pointage sémantique. L'ordre des exercices a été contrebalancé entre les participants. Pour les 2 exercices, les participants devaient saisir 21 phrases, choisies aléatoirement dans un corpus de 50 phrases.

A partir des données recueillies, nous avons calculé la vitesse de saisie de texte pour chaque mot saisi. Le temps de saisie a été calculé entre le premier caractère du mot et l'appui sur la barre espace à la fin du mot. La vitesse de saisie était de 13,10 mots par minutes pour le clavier sémantique, contre 11,54 mots par minutes avec le clavier AZERTY, soit un gain de 14 %, statistiquement significatif ($F_{1,10} = 21,12$ et $p < 0,01$). Ce gain de vitesse avec le clavier sémantique a pu être constaté dès la première phrase saisie et n'a pas beaucoup évolué au cours de l'exercice (cf. Figure 17)

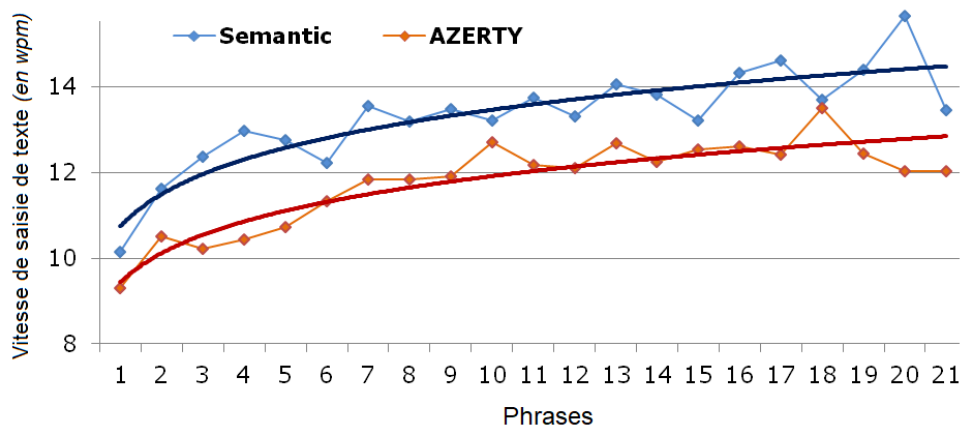


FIGURE 17 – Vitesse de saisie de texte (en mots par minute) en fonction du nombre de phrases saisies et du clavier utilisé

Ces premiers résultats ont été renforcés par une seconde expérimentation avec un participant souffrant d'une dystrophie musculaire. Ce dernier a réalisé la même expérience que les participants valides, mais avec moins de phrases à copier pour anticiper de potentiels problèmes de fatigue musculaire. Les résultats obtenus lors de cette expérimentation complémentaire sont du même ordre que ceux obtenus lors de la première expérimentation : la vitesse de saisie de texte avec les 2 claviers est plus basse, mais l'écart entre les deux est identique : 7,59 mots par minute avec le clavier sémantique contre 6,58 mots par minute avec le clavier AZERTY soit 15% d'augmentation en faveur du clavier sémantique.

Ces résultats ont donc validé notre hypothèse que sans modification de l'aspect graphique, l'utilisateur n'est pas perturbé dans sa saisie par les éléments modifiés. Le pointage sémantique permet cela en facilitant la tâche de pointage sans pour autant modifier l'apparence des touches.

Ce travail a été le fruit d'une collaboration avec Scott Mackenzie de l'université de York (Toronto, Canada) et de Bruno Merlin, alors en poste d'enseignant chercheur à l'université fédérale do Para (Cameta, Brésil). Les résultats plus détaillés de cette étude ont été publiés à la conférence ICCHP 2014 [[RAYNAL et al., 2014](#)].

C Utilisation de la déduction de mots

Mes derniers travaux sur la saisie de texte avaient un contexte différent des précédents travaux. Ces derniers ont porté sur l'accessibilité de la saisie de texte sur smartphone pour des personnes déficientes visuelles. Cette étude était l'objet des travaux de thèse de Philippe Roussille que j'ai encadré de 2013 à 2017.

L'enjeu était de permettre à un utilisateur non voyant d'utiliser un clavier logiciel traditionnel sur un écran tactile. Sans repères tactiles, les utilisateurs non-voyants ne peuvent pas prendre connaissance de la position exacte de chaque touche affichée à l'écran. La solution généralement utilisée consiste à naviguer avec le doigt sur l'écran tactile. Un retour vocal est alors donné pour chaque caractère survolé par le doigt. Cette phase de navigation est très coûteuse en temps. Par conséquent, la vitesse de saisie de texte est réduite par rapport à l'utilisation d'un clavier logiciel où l'utilisateur tape directement sur les touches.

Nous sommes partis du constat que l'utilisateur non-voyant a une très bonne connaissance de la disposition des caractères sur son clavier physique. D'autre part, celui-ci n'a pas besoin d'avoir un retour pour chaque caractère sélectionné. Un retour vocal à chaque fin de mot est suffisant. Notre idée était donc d'utiliser un algorithme de déduction de mots qui serait capable de déduire le mot souhaité par l'utilisateur à partir d'une saisie approximative sur le clavier. Le principe du clavier DUCK (pour DedUctive Keyboard) est donc le suivant :

le clavier logiciel est affiché sur toute la surface de l'écran du smartphone. Afin de ne pas perdre la connaissance préalable de l'agencement du clavier, l'emplacement des caractères et des touches les plus importantes est identique à ceux de l'agencement du clavier de l'utilisateur. Lorsque l'utilisateur souhaite saisir un mot, il navigue d'abord avec un doigt sur le clavier afin de chercher la première lettre. Au cours de cette exploration, l'utilisateur entend un retour vocal correspondant au caractère touché. Pour valider la première lettre, il relève le doigt de l'écran. Pour saisir le reste des caractères du mot à saisir, l'utilisateur frappe sur le clavier logiciel à la position où il pense trouver le caractère voulu. L'utilisateur doit effectuer ces frappes sur le clavier sans se soucier de la précision spatiale. Afin de ne pas perturber l'utilisateur, il n'y a pas de retour vocal sur ces frappes, mais seulement un retour audio simple (bip) indiquant que la frappe a bien été prise en compte. Une fois que l'utilisateur a fini d'effectuer les frappes correspondantes au reste des caractères, il valide sa saisie en effectuant un appui à deux doigts sur l'écran tactile. En fonction de la première lettre validée, du nombre de frappes et de leur position, l'algorithme de déduction détermine les mots qui correspondent le mieux à cette séquence, et renvoie la liste des 4 mots les plus proches. Le premier mot est lu et épelé. L'utilisateur peut sélectionner le mot en cours en appuyant sur deux doigts, ou faire défiler la liste avec un doigt. En cas d'erreur, l'utilisateur peut annuler toute entrée de texte en faisant glisser deux doigts vers la gauche.

Les premiers résultats d'évaluation ont montré que cette technique était avantageuse si le mot à saisir comportait au moins 5 caractères [ROUSSILLE et al., 2015]. En revanche pour les mots de moins de 5 caractères, la phase de validation du mot à la fin était trop coûteuse en temps pour que la saisie sur le clavier DUCK soit plus efficace que sur un clavier classique où l'utilisateur saisit chaque caractère après avoir navigué sur le clavier. C'est pourquoi, au-delà de la technique d'interaction utilisée pour saisir un mot et de l'algorithme de déduction proposée, nous nous sommes également intéressés à la saisie de mots courts [ROUSSILLE et RAYNAL, 2016b] et à la sélection des mots dans la liste de mots proposée [ROUSSILLE et RAYNAL, 2016a].

Le nouveau prototype du clavier DUCK, intégrant les résultats des travaux sur les mots courts et sur l'interaction avec la liste de mots, a finalement été évalué au travers d'une étude longitudinale [RAYNAL et ROUSSILLE, 2018]. Cette étude a été menée auprès de 4 personnes non voyantes. Afin d'évaluer l'évolution des performances des participants lors des premières utilisations du clavier DUCK, nous avons réalisé deux sessions d'évaluation, séparées de deux semaines. Après la première session, le clavier DUCK a été laissé aux participants pendant deux semaines. Nous leur avons demandé d'utiliser quotidiennement le clavier, pour réaliser une production écrite correspondant à quelques lignes de texte, comme un email. Pour les deux sessions, les participants devaient saisir 20 phrases avec respectivement le clavier DUCK et le clavier VODKA. Ce dernier est une reproduction d'un clavier AZERTY avec sélec-

tion des caractères en navigant avec le doigt sur le clavier. L'ordre des exercices sur les deux claviers a été contrebalancé entre les participants.

La Figure 18 donne la vitesse de saisie moyenne obtenue lors des deux sessions pour les deux claviers. Le graphique de la Figure 1 donne les vitesses de saisie de texte par session et par clavier. On peut constater que la vitesse moyenne de saisie de texte est déjà supérieure avec le clavier DUCK avec 0,71 caractères saisis par seconde (cps), soit 8,52 mpm, par rapport au clavier VODKA (0,56 cps / 6,72 mpm). Cet écart s'accroît lors de la deuxième session où les utilisateurs ont atteint une vitesse de saisie de 1,1 cps (13,2 mpm) avec le clavier DUCK, contre une moyenne de 0,61 cps (7,32 mpm) avec le clavier VODKA.

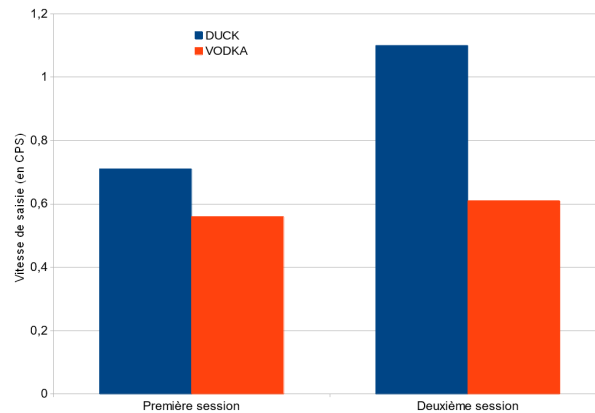


FIGURE 18 – Vitesse de saisie avec les deux claviers à l'issue de la première et de la deuxième session

D'autre part, nous pouvons constater sur la Figure 19 que lors de la deuxième session, les participants ont saisi globalement plus rapidement avec le clavier DUCK qu'avec le clavier VODKA, quelle que soit la longueur du mot. Plus la longueur du mot augmente et plus le système DUCK augmente la vitesse de saisie de texte.

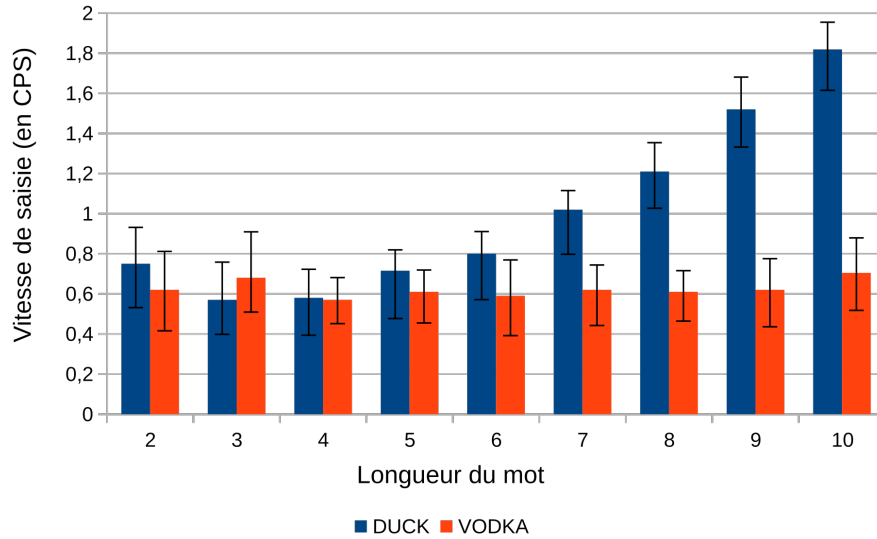


FIGURE 19 – Vitesse de saisie avec les deux claviers à l'issue de la deuxième session

D Synthèse

Dans le chapitre précédent, l'étude sur l'utilisation de la liste de mots a montré l'importance de l'attention de l'utilisateur sur sa tâche de pointage et l'impact qu'avaient des modifications dynamiques en cours de saisie.

Une partie des travaux présentés dans ce chapitre confirment ce constat : les modifications dynamiques engendrent une perte d'attention de l'utilisateur sur sa tâche principale. Malgré la diminution des distances à parcourir, les systèmes KeyGlass et FloodKey ne permettent pas de saisir plus rapidement qu'avec un clavier logiciel traditionnel. De même, bien que les algorithmes de prédiction de mots permettent de réduire le nombre d'actions à effectuer pour saisir un mot, cela ne suffit pas toujours pour que la vitesse de saisie augmente en utilisant ces listes.

L'enjeu de mes recherches était donc d'étudier différentes stratégies qui pourraient tirer profit des résultats des algorithmes de prédiction (de mots ou caractères) pour faciliter la saisie de l'utilisateur et que celui-ci en tire un bénéfice par rapport à la saisie sur un clavier logiciel traditionnel. Les principaux résultats obtenus l'ont été au cours des travaux de thèse de 3 étudiants que j'ai encadrés.

D'une part, Georges Badr, que j'ai encadré de 2007 à 2011, a contribué à l'amélioration des listes de mots proposées en complément des claviers logiciels. Le système WordTree [BADR et RAYNAL, 2009a; BADR et RAYNAL, 2009c; BADR et RAYNAL, 2010] utilisé avec un clavier logiciel, puis la disposition CentraList [BADR et RAYNAL, 2011] qui met au centre du système de saisie la liste de mots permettent à l'utilisateur d'améliorer ses performances de saisie.

De même, Bruno Merlin, que j’ai encadré de 2008 à 2011, a conçu le système SpreadKey [MERLIN et RAYNAL, 2009; MERLIN et RAYNAL, 2010] qui utilise les résultats d’un algorithme de prédiction de caractères pour répandre les caractères les plus probables sur le clavier. Notre collaboration avec Scott Mackenzie de 2012 à 2014 nous a également permis de mettre en œuvre le clavier sémantique [RAYNAL et al., 2014] qui facilite la tâche de pointage sur le clavier sans modifier l’aspect graphique du clavier.

Enfin, Philippe Roussille, dont j’ai encadré les travaux de recherche durant son Diplôme Universitaire (2012-2013) puis tout au long de son doctorat (2013-2017), a étudié l’intégration des algorithmes de déduction dans les claviers logiciels utilisés sur écran tactile afin de permettre leur utilisation par des utilisateurs non-voyants. Ces travaux ont notamment abouti à la création du clavier DUCK [ROUSSILLE et al., 2015; RAYNAL et ROUSSILLE, 2017; RAYNAL et ROUSSILLE, 2018].

Au travers de ces travaux, nous avons également pu constater à quel point il était difficile de modifier le clavier sans que cela ne perturbe l’utilisateur. Afin de limiter les perturbations que cela occasionne, nous avons également étudié la transition entre deux dispositions de caractères [MERLIN et al., 2012b; MERLIN et al., 2012c] en reprenant le concept de multi-layer introduit dans [MERLIN et al., 2009]. Lors d’une collaboration avec Benoît Martin, nous avons également étudié la prévisualisation des résultats de la prédiction de caractères en affichant les caractères à venir en fonction du caractère que l’utilisateur survole. Nous avons ainsi pu constater que la prévisualisation permet à l’utilisateur d’anticiper la sélection du caractère suivant et par conséquent de gagner du temps lors de la sélection du caractère suivant [GODARD et al., 2009; RAYNAL et MARTIN, 2020].

Deuxième partie

Dispositifs et techniques de pointage

Introduction

Depuis l'apparition à la fin des années 70 des interfaces dites WIMP (Acronyme anglais pour « Windows, Icons, Menus and Pointing device »), le pointage est devenu une activité essentielle en informatique. En effet, quel que soit le dispositif de pointage et le support, on utilise régulièrement le pointage pour des tâches de sélection ou de désignation d'objets. Malgré l'évolution des interfaces graphiques avec notamment les interfaces post-WIMP [[DAM, 2001](#)], la tâche de pointage continue d'être prépondérante pour les utilisateurs.

Avant même les études de son utilisation en informatique, la tâche de pointage était déjà l'objet de nombreuses recherches dans le monde physique [[WOODWORTH, 1899](#)]. L'étude la plus reconnue est celle menée par Fitts au début des années 1950. La loi de Fitts permet de prédire le temps moyen dont on a besoin pour réaliser une tâche de pointage en fonction de la distance à parcourir pour atteindre la cible et de la taille de la cible à pointer.

En informatique, cette loi est aussi utilisée pour évaluer et comparer *a priori* l'utilisabilité d'interfaces graphiques ou de mesurer les performances de nouveaux périphériques de pointage. Depuis le début des années 2000, les surfaces d'affichage et les dispositifs de pointage ont beaucoup évolué, de nouveaux ont émergé. Les écrans fixes utilisés avec ordinateur de bureau laissent de plus en plus de place à différentes surfaces d'affichage soit réduites (écran de tablettes, smartphone, ou encore montres connectées), soit, au contraire, très largement agrandies tels que les murs d'écran [[NANCEL et al., 2013](#)]. De nouveaux dispositifs d'affichage sont également apparus comme, par exemple, les lunettes type Hololens.

De la même manière, les dispositifs de pointage ont aussi évolué. La simple souris manipulée sur un plan en 2 dimensions laisse régulièrement la place à des dispositifs plus complexes manipulables dans l'espace physique et laissant ainsi plus de possibilités à l'utilisateur : Wii, Kinect, Gant, Smartphone, capteurs...

D'autre part, de plus en plus de données sont mises à disposition des utilisateurs. Cette multitude de données recueillies nécessite d'être présentée à l'utilisateur de manière à ce que

celui-ci puisse en prendre connaissance et les analyser. Différentes formes de représentations sont utilisées pour permettre une meilleure visualisation : en fonction du type de données présentées, celles-ci peuvent être affichées sous forme de listes, de tableaux, d'arbres, de graphes, ou encore de structures plus complexes utilisant des représentations en 3 dimensions. Au-delà de la visualisation de ces données, il est également nécessaire de pouvoir manipuler et en premier lieu de pointer les données souhaitées pour pouvoir ensuite les manipuler.

Quel que soit le contexte d'interaction obtenu par la combinaison de ces 3 composantes de l'environnement (dispositifs de pointage, surfaces d'affichage et types de représentation des données), la loi de Fitts fait office de référence pour l'évaluation de la tâche de pointage. Mes premiers travaux sur le pointage m'ont donc amené à m'intéresser à la loi de Fitts et proposer une manière de mieux prendre en compte la variabilité des performances des utilisateurs pour la tâche de pointage dans un contexte d'interaction donné.

Mes travaux se sont ensuite axés sur la tâche de pointage sur de grandes surfaces distantes, d'une part, et dans des environnements virtuels 3D d'autre part. Ces deux contextes présentent des problématiques différentes. Dans le cas des grandes surfaces, il s'agit d'améliorer le déplacement du pointeur sur de grandes distances, ce que ne permettent pas les dispositifs actuels. Dans le cas des environnements virtuels 3D, il s'agit surtout d'appréhender les problèmes liés à la profondeur.

De la même manière que pour les systèmes de saisie de texte, mes contributions sont de deux types, et seront donc présentées au travers de deux chapitres présentant les principaux travaux.

Le premier chapitre présente les contributions méthodologiques. Hormis l'étude sur la variabilité des performances de pointage, mes contributions théoriques se sont axées autour de la tâche de pointage dans un environnement virtuel 3D (EV 3D). Nous nous sommes intéressés d'une part à la méthodologie d'évaluation des dispositifs de pointage pour une tâche de pointage dans un EV 3D. La méthode et les métriques d'évaluation des dispositifs de pointage pour une tâche de pointage dans une interface à deux dimensions sont normalisées depuis une dizaine d'année. En revanche cette norme ne préconisait rien pour l'évaluation dans des environnements virtuels 3D. Ces environnements étant de plus en plus utilisés, nous nous sommes intéressés à l'adaptation de cette norme pour les EV 3D. D'autre part, la conception de techniques d'interaction avancées et la conception d'environnements virtuels 3D sont 2 domaines de recherche spécifiques qui sont rarement traités ensemble pour apporter les techniques d'interaction les plus appropriées aux tâches relatives à la manipulation d'éléments dans un EV 3D. C'est pourquoi nous nous sommes également intéressés à la définition d'un espace de conception prenant en compte à la fois les préoccupations liées à l'IHM et celles plus spécifiques liées à la synthèse d'image.

Le second chapitre traite des différentes techniques d'interaction avancées que nous avons mises en œuvre pour répondre à la problématique du pointage sur de grandes surfaces d'affichage, dans des environnements virtuels 3D ou dans des contextes d'interaction où la représentation des données est hétérogène.

Chapitre 4

Aide à la conception et à l'évaluation de techniques de pointage

Mes travaux théoriques sur la tâche de pointage se sont décomposés en 3 axes de recherche. Mon premier travail s'est focalisé sur le modèle de prédiction du temps de pointage. Notre objectif était de proposer une adaptation de la loi de Fitts pour prendre en compte la variabilité du mouvement et, par conséquent, du temps de pointage nécessaire pour réaliser une tâche de pointage quel que soit le contexte d'interaction (pointage 2D, dans un environnement 3D, avec ou sans déficience motrice).

Pour le second axe de travail, nous nous sommes focalisés sur la tâche de pointage dans un environnement virtuel 3D. L'enjeu était d'adapter la norme d'évaluation existante pour les tâches de pointage dans un plan 2D à celles effectuées dans un environnement 3D. Pour faciliter sa mise en œuvre, nous avons également proposé une plate-forme d'évaluation des dispositifs de pointage pour les tâches de pointage dans un EV 3D.

Enfin, le dernier travail théorique focalisait également sur les environnements virtuels 3D, et prenait en compte toutes les tâches d'interaction. Nous nous sommes intéressés à mettre en œuvre un cadre de conception permettant d'associer les tâches d'interaction de l'utilisateur aux différents éléments composant l'environnement virtuel 3D.

A Évaluation de la tâche de pointage : Fuzzy Fitts

La tâche de pointage fait l'objet de nombreuses études, dont celles réalisées par Paul Fitts qui servent de référence depuis maintenant près de 70 ans [FITTS, 1954; FITTS et PETERSON, 1964]. Ses études ont été effectuées dans un environnement physique pour une tâche de pointage sur une seule dimension, et ont abouti à la proposition d'un modèle prédictif défini comme tel :

$$MT = a + b \times \log(ID) \quad (4.1)$$

Où ID est l'indice de difficulté de la tâche de pointage à effectuer et est dépendant de la taille de la cible à atteindre et de la distance à parcourir pour l'atteindre. Au fil du temps et de multiples études, il a été démontré que ce modèle prédictif pouvait être utilisé pour des pointages dans des environnements réels mais également s'appliquer à des pointages dans des environnements numériques [CARD et al., 1978], effectués avec tout type de dispositifs de pointage, et quel que soit le nombre de dimensions utilisées [MACKENZIE et BUXTON, 1992]. En effet, quel que soit le contexte d'interaction, le modèle proposé par Fitts peut être adapté en recalculant les valeurs de a et b.

Les valeurs pour a et b sont obtenues à partir de l'équation de la droite de régression linéaire obtenue sur un ensemble de données de pointage effectuées pour plusieurs tâches de pointage avec divers indices de difficulté. L'utilisation de la régression linéaire représente dans ce modèle un biais car cela suppose que les données sont linéaires, ce qui est rarement le cas. Ainsi, la fonction de régression utilisée ne rend pas compte de l'imprécision qui est induite par le modèle et les données. D'autre part, la régression linéaire permet difficilement d'analyser la variabilité de performances lors de la tâche de pointage. Les calculs d'intervalle de confiance pour la pente et l'ordonnée à l'origine permettent de décrire dans quelle mesure le modèle appris peut varier du concept cible. Cependant, les calculs d'intervalle de confiance se font a posteriori et sont biaisés par le modèle issu de la régression : les droites de confiance sont parallèles et symétriques autour de la droite de régression. Elles ne peuvent donc pas mettre en évidence les différences de dispersion entre un indice de difficulté faible et un autre élevé. D'autre part, il est possible de faire une analyse de dispersion permettant une étude locale de la variation uniquement sur les indices de difficulté pour lesquels nous avons des données issues de l'expérience. Ces analyses ne sont pas utilisables pour les indices de difficulté non présents dans l'expérience. On ne peut donc pas les utiliser pour prédire la variabilité sur une tâche de pointage avec un indice de difficulté quelconque.

C'est pourquoi nous nous sommes intéressés à un modèle de données moins précis. La théorie des ensembles flous a été introduite en 1965 par Zadeh [ZADEH, 1965] afin de représenter de manière plus réaliste les classes d'objets qui ne possèdent généralement pas de critères d'appartenance bien tranchés. L'idée de Zadeh consiste à ne pas définir de seuil, mais plutôt à autoriser des degrés d'appartenance compris entre 0 et 1. Ce type d'ensemble se représente par un trapèze. Pour notre cas d'utilisation (la loi de Fitts), les ensembles flous déterminent la plausibilité d'un temps pour un indice de difficulté donné. Ainsi l'axe des abscisses représente les valeurs de temps et celui des ordonnées reste le taux de plausibilité (cf. Figure 20).

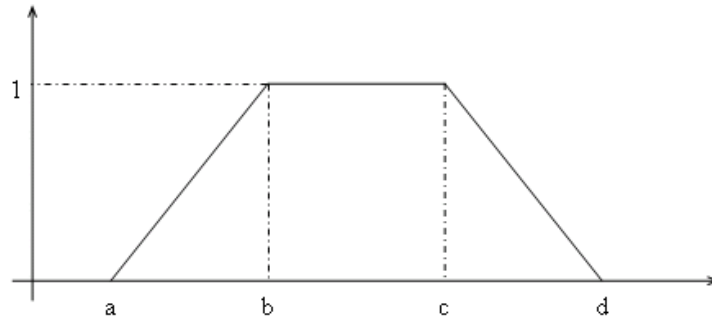


FIGURE 20 – Vitesse de saisie de texte (en mots par minute) en fonction du nombre de phrases saisies et du clavier utilisé

Le but de la régression floue est de dépasser ces limitations en apprenant non plus des fonctions de régression précises (dans le cas de la tâche de pointage, il s'agit de données temporelles liées à la difficulté de la tâche), mais plutôt des fonctions imprécises, dont l'imprécision est représentée par des ensembles flous. Nous avons instancié le modèle de la loi de Fitts sur deux types d'approche utilisant la régression floue. La première est la régression floue [DIAMOND, 1988] à partir de données elles-mêmes floues. La seconde approche décrit la régression floue à partir de données classiques [TANAKA, 1987]. Cette méthode est nommée régression imprécise. Dans ce type d'approche, la fonction de régression associe à un vecteur de variables d'entrée un ensemble flou décrivant les valeurs possibles pour la variable de sortie. Ces valeurs vont de 0, valeur de sortie impossible, à 1 pour une valeur de sortie complètement plausible. On obtient ainsi une description de la tendance générale des données et de la variation autour de cette tendance. On utilise la méthode décrite dans [SERRURIER et PRADE, 2006], car elle permet de représenter l'imprécision avec des ensembles flous trapézoïdaux et est robuste en présence de données incohérentes.

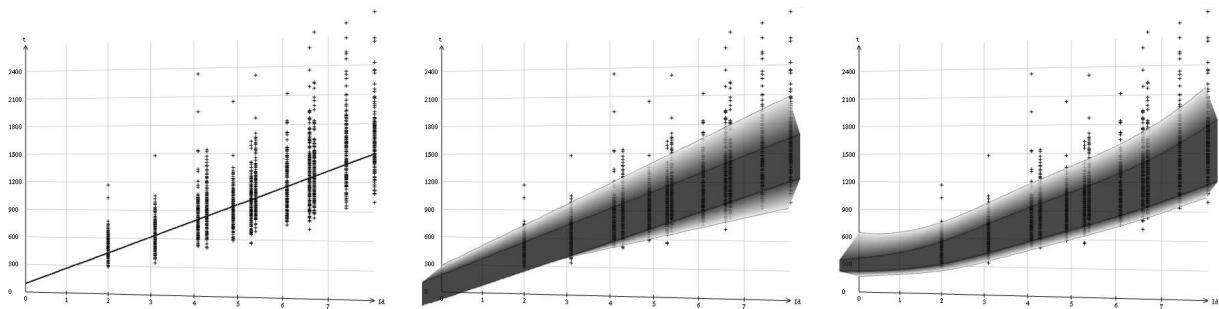


FIGURE 21 – régression linéaire classique; régression floue linéaire; régression floue non linéaire

Avec la régression floue, on fait l'hypothèse que le concept que l'on veut décrire est naturellement imprécis : pour un même indice de difficulté, deux tâches de pointage ne se réalisent pas forcément avec exactement le même temps. Dans le cas de Fitts, cela revient à faire l'hypothèse que pour un indice de difficulté donné, les temps de mouvement possibles peuvent

être décrits sous la forme d'un ensemble flou. Ainsi la dispersion autour de la tendance générale est prise en compte à tous les niveaux de l'apprentissage. Il ne s'agit donc plus, comme dans le cas de la régression linéaire, de déterminer la tendance centrale des données, mais de décrire de la manière la plus fiable possible toutes les données. Le modèle obtenu à l'aide de la régression floue permet d'obtenir directement la plausibilité des temps de pointage pour un indice de difficulté. Ce dernier peut être différent de ceux utilisés lors de l'expérience. La régression floue permet de décrire de manière plus exhaustive les temps de mouvement possibles pour un indice de difficulté donné. De plus elle prend en compte et représente directement la dispersion des données pour tout indice de difficulté.

La régression floue peut avoir des avantages pour deux types d'analyses régulièrement effectuées lorsqu'on évalue la tâche de pointage. La première est l'analyse de la courbe elle-même afin d'en déduire des informations sur la nature psychomotrice de la tâche effectuée. Cela permet entre autres de comparer les différences de performances de plusieurs techniques ou dispositifs de pointage. Dans ce cas, la régression floue peut permettre de faire ressortir des phénomènes qui ne sont pas immédiatement détectables lorsque l'on utilise la régression classique. Par exemple, si la dispersion augmente beaucoup avec la difficulté, cela peut signifier que d'autres paramètres que la difficulté de pointage interviennent dans le processus d'accomplissement de la tâche. Au contraire, si la dispersion reste stable et faible, on peut considérer que la difficulté est la seule variable significative pour déterminer le temps de réalisation de l'action.

La seconde est l'utilisation de la loi de Fitts pour prédire le temps de réalisation d'une tâche (ou d'une succession de tâches) afin de pouvoir, par exemple, optimiser la conception des interfaces Homme Machine. Dans ce cas, la régression floue permet des analyses beaucoup plus fines des résultats grâce à l'arithmétique sur les ensembles flous (addition et comparaison). On peut ainsi connaître le temps moyen pour la résolution d'une tâche, mais aussi les temps extrêmes et la répartition des valeurs de temps dans cet intervalle. De plus, dans le cas où deux valeurs de temps sont très proches avec l'approche classique, la comparaison des ensembles flous permet des résultats moins ambigus.

La régression floue peut ainsi être considérée comme un outil d'analyse complémentaire à la régression linéaire classique dans le cadre de l'apprentissage de la loi de Fitts. La régression linéaire permet de définir une loi simple à partir de laquelle on peut obtenir le temps moyen de mouvement pour un indice de difficulté, alors que la régression floue permettra d'obtenir des informations plus précises sur la variabilité des données en fonction de la difficulté de la tâche. Ces travaux sur la régression floue sont plus difficilement utilisables à grande échelle car la régression floue est beaucoup plus difficile à mettre en œuvre par rapport à la régression linéaire qui peut être calculée avec un simple tableur.

Ce travail a été réalisé en collaboration avec Mathieu Serrurier, de l'équipe ADRIA de l'IRIT, qui est enseignant chercheur en Intelligence Artificielle, spécialisé en Apprentissage. Notre travail sur la régression floue appliquée au modèle prédictif de Fitts est à la frontière de nos deux disciplines et les résultats ont été présentés dans une conférence de nos domaines respectifs. L'article présenté à la conférence IHM met en avant les scénarii d'utilisation de la régression floue dans le domaine de l'IHM et plus particulièrement l'aide qu'il peut apporter sur l'évaluation de la tâche de pointage et l'aide à la conception des futures interfaces [RAYNAL et SERRURIER, 2007]. L'article publié à la conférence internationale NAFIPS présente plus en détail la manière de calculer les ensembles flous et de réaliser la régression sur ces données floues [SERRURIER et RAYNAL, 2008].

B Évaluation du pointage pour les environnements 3D

Bien que la loi de Fitts présente un certain nombre d'inconvénients (présentés dans la section précédente), le modèle prédictif de la loi de Fitts fait office de référence pour l'évaluation d'une tâche de pointage. La norme ISO 9241-9 [ISO, 2000] donne même un ensemble de recommandations pour mettre en œuvre l'évaluation de dispositifs de pointage tels que la souris, le joystick, etc. Cette norme a pour vocation d'uniformiser la manière d'évaluer les dispositifs de pointage au travers d'une tâche de pointage standardisée. D'autre part, la norme décrit également les métriques à utiliser pour évaluer les performances des dispositifs. En respectant les recommandations de cette norme, les chercheurs assurent plus de robustesse pour les données recueillies. Ces recommandations permettent également de faciliter la cohérence des données d'une étude à l'autre et par conséquent, améliore la comparabilité des études entre elles [R. W. SOUKOREFF et MACKENZIE, 2004].

Cependant, cette norme, réalisée au début des années 2000, définit la tâche de pointage dans un plan en deux dimensions et ne prend donc en considération que les tâches de pointage en 2D. Des études ont depuis repris les recommandations de cette norme pour les adapter à des tâches en 3D [TEATHER et STUERZLINGER, 2013]. Cependant la 3D y est souvent considérée comme des éléments 3D présentés sur un plan (X, Y) avec des hauteurs différentes sur l'axe Z (cf. Figure 22).

Notre travail a porté sur la tâche de pointage dans de « véritables » environnements 3D où chaque objet peut être disposé n'importe où dans la scène. L'objectif de ce travail était d'adapter la norme ISO existante aux environnements virtuels 3D, en s'assurant que les caractéristiques et contraintes liées à un EV 3D soient bien prises en compte dans cette adaptation de la norme. Au-delà de l'adaptation à l'EV 3D, des recommandations et des mesures proposées dans la norme, nous avons également conçu et développé une plate-forme d'évaluation

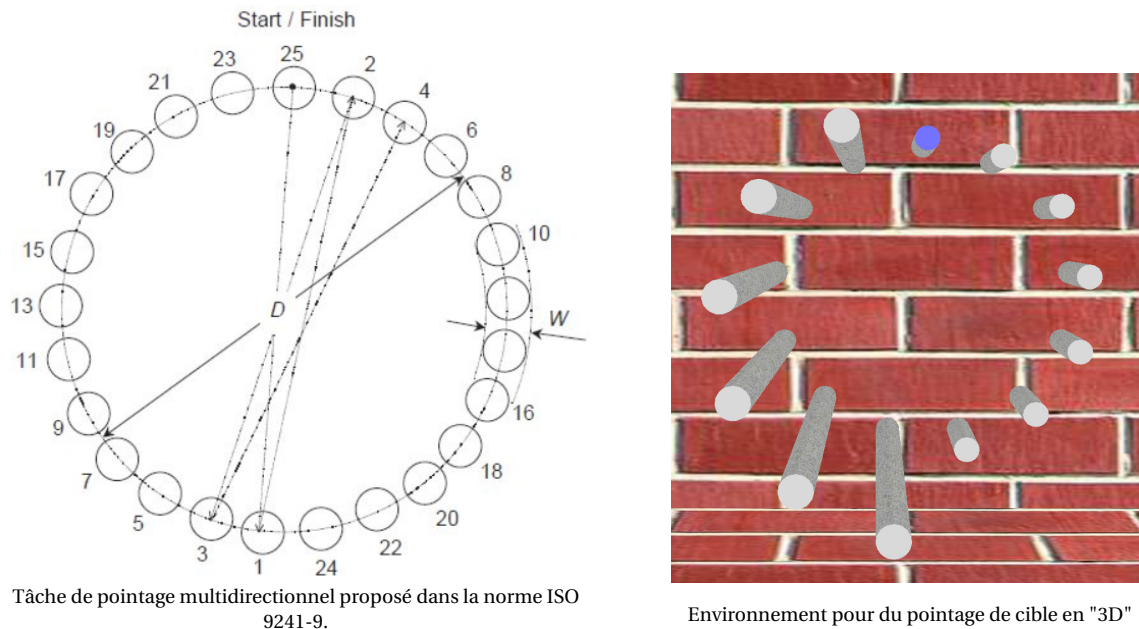


FIGURE 22 – Vitesse de saisie avec les deux claviers

permettant d'évaluer les dispositifs de pointage sur la tâche de pointage en 3D que nous recommandons.

1 Adaptation de la norme d'évaluation

Comme nous l'avons vu précédemment, la loi de Fitts est utilisée pour prédire le temps moyen nécessaire pour effectuer une tâche de pointage. La norme ISO 9241-9 [ISO, 2000] fait maintenant office de référence pour mettre en œuvre l'évaluation d'un dispositif de pointage. Les recommandations de cette norme servent à mettre en œuvre une expérimentation pour évaluer la tâche de pointage dans un contexte d'interaction particulier. Les données recueillies lors de cette expérimentation permettent de calculer la droite de régression linéaire qui nous donne les coefficients a et b du modèle de Fitts (cf. section précédente). En plus de paramétrer le modèle prédictif pour un contexte d'interaction spécifique, cette expérimentation permet également d'évaluer les performances d'un dispositif de pointage [R. W. SOUKOREFF et MACKENZIE, 2004]. En suivant les mêmes recommandations, les performances estimées pour chaque dispositif de pointage sont plus facilement comparables les unes aux autres. La norme ISO définit principalement 3 aspects pour évaluer la tâche de pointage : la tâche à réaliser, les caractéristiques des cibles à utiliser, et les métriques à calculer. Pour notre adaptation de la norme ISO aux environnements virtuels 3D, nous nous sommes donc attachés à adapter ces 3 aspects aux spécificités de la 3D.

La **tâche** recommandée par la norme ISO consiste à sélectionner une cible fixe aussi rapidement et précisément que possible. La sélection comprend le pointage d'une cible en déplaçant un curseur d'une cible de départ à une cible de destination avec le dispositif de pointage. La tâche de pointage se termine par la validation de la sélection au moyen d'un appui sur un contacteur (généralement un bouton du dispositif de pointage). Si le curseur est à l'intérieur de la cible de destination lorsque l'utilisateur confirme la sélection, la tâche de pointage est réussie. Si le curseur est en dehors de la cible, la validation génère une erreur. Dans tous les cas, l'appui sur le bouton ou la touche marque la fin de la tâche en cours et le début de l'essai suivant dans le protocole. La difficulté de la tâche varie en fonction des indices de difficulté. Ces indices sont dépendants de la taille de la cible et de la distance à parcourir. Ces deux paramètres varient au cours de l'expérimentation : chaque série de tâches est une nouvelle combinaison entre taille et distance de manière à avoir des ID différents. Dans notre adaptation, l'instruction de la tâche, l'existence d'une cible source et d'une cible destination et la nécessité d'une confirmation ont été conservées. L'adaptation concerne d'abord la validation. La norme ISO ne recommande aucun élément de représentation spécifique. Malgré cela, le pointeur est couramment représenté par une croix. Dans notre adaptation, la représentation volumétrique du curseur que nous recommandons est une sphère. De plus, sa largeur recommandée est d'environ 70% de la plus petite largeur de la cible pour rester visible quelle que soit sa position dans l'environnement 3D. La forme du curseur est définie afin de s'assurer que le point de collision est capturé à une même distance du centre du curseur au centre de la cible quel que soit le mouvement 3D (ce qui n'est pas possible avec un carré). Au moment de la validation, il n'est pas nécessaire que ce pointeur 3D soit complètement à l'intérieur de la cible pour que la validation soit considérée comme une sélection réussie. La détection d'une collision entre la cible et le curseur est suffisante.

Les **cibles** ne sont pas explicitement définies par la norme, mais les cibles circulaires sont recommandées et sont les plus utilisées dans les évaluations en 2D, car un cercle a une largeur unique (son diamètre) quel que soit le mouvement de l'utilisateur. Par conséquent, notre adaptation à la 3D recommande que les cibles deviennent des sphères afin de conserver une seule mesure pour la largeur indépendamment du mouvement 3D : la taille de la cible sera donc donnée par le diamètre des sphères. De la même manière, la norme ISO recommande d'espacer les cibles autour de la circonférence d'un cercle. En outre, l'emplacement des cibles sur le cercle doit être défini de manière à ce que chaque cible de départ d'une tâche soit symétrique, par rapport au centre du cercle support, à la cible à atteindre. Cela permet de garantir que la distance entre deux cibles est toujours la même et qu'elle est égale au diamètre du cercle support. Dans notre adaptation, le cercle support est donc remplacé par une sphère support (cf. Figure 23). Nous conservons les mêmes contraintes pour le positionnement des cibles sur cette sphère support.

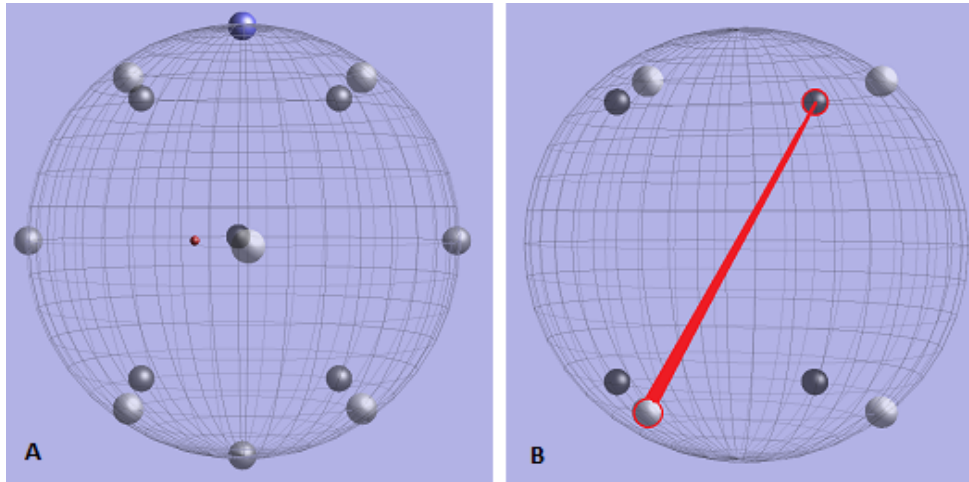


FIGURE 23 – Répartition des cibles et exemple de tâche de pointage

Enfin, la norme ISO recommande différentes mesures dont le débit et le temps de mouvement moyen. Dans notre adaptation, ces mesures ont été conservées. Pour pouvoir les calculer, la norme définit également d'utiliser comme variables indépendantes la taille des cibles et la distance qui les sépare afin de pouvoir obtenir différents indices de difficulté. Le calcul de l'indice de difficulté a été modifié dans notre adaptation de manière à pouvoir prendre en compte le volume du pointeur 3D. Son calcul, comme illustré dans l'équation ci-dessous, inclut donc la taille du curseur :

$$ID = \log_2 \left(\frac{D}{W_{cible} + W_{curseur}} + 1 \right) \quad (4.2)$$

Le calcul des différentes distances est obtenu en prenant le centre des sphères comme point de référence.

2 Plate-forme d'évaluation pour les pointages en 3D

Pour faciliter l'utilisation des recommandations que nous préconisons, nous avons développé une plate-forme d'évaluation qui met en œuvre un environnement virtuel 3D respectant les caractéristiques de l'environnement que nous avons défini pour l'évaluation de la tâche de pointage (cf. Figure 24). Cette plate-forme a pour but de soutenir le déploiement du protocole afin d'évaluer les techniques de saisie 3D et de collecter les données issues de l'expérimentation. Elle est composée d'un environnement 3D et d'un panneau de configuration. La tâche de pointage est composée d'un certain nombre de cibles que l'expérimentateur peut gérer avec le panneau de configuration : les cibles peuvent être placées à des longitudes et latitudes spécifiques. Un bloc est composé d'une série de tâches de pointage, qui ont le même

ID. L'expérimentateur peut configurer la taille des cibles (largeur) et la taille du support (distance) pour ajuster l'ID du bloc.

Pendant l'expérimentation, la plate-forme enregistre tous les événements relatifs à la tâche tels que le début et la fin de chaque tâche de pointage, la validation de la sélection et le temps d'exécution. De même, elle enregistre la position du curseur 3D et les erreurs lorsque la validation a été effectuée sans collision entre le curseur et la cible. Tous les événements de l'expérience sont datés et enregistrés dans un format XML.

Nous avons notamment utilisé cette plate-forme dans le cadre de l'évaluation d'une technique d'interaction ubiquitaire [SCHMITT, DUBOIS et al., 2011].

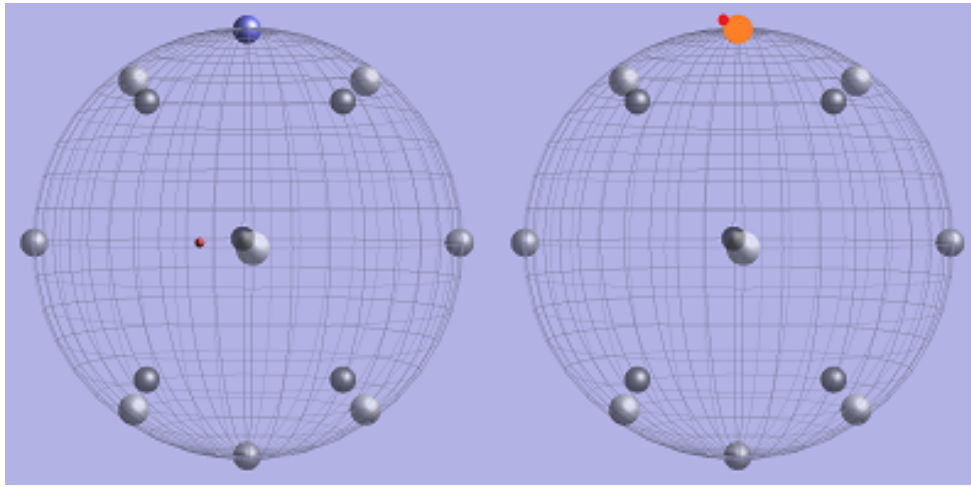


FIGURE 24 – Plateforme pour les tâches de pointage en 3D

Lors de la conception de cette plate-forme, nous avons également porté une attention toute particulière à la perception de la profondeur dans l'EV 3D. En effet, en fonction du matériel utilisé, les utilisateurs peuvent ne pas être en mesure de percevoir correctement la profondeur d'un objet au sein de la scène. C'est pourquoi nous avons utilisé certains indices visuels 3D pour surmonter ces difficultés de perception de la profondeur : par exemple, nous utilisons une graduation de couleurs combinée à une transparence des sphères en fonction de leur profondeur dans la scène par rapport au point de vue de l'utilisateur [ZHAI et al., 1994]. Cette transparence réduit l'occlusion avec les autres sphères et permet aux utilisateurs de voir le curseur derrière les sphères. Une taille relative des sphères est due à la perspective de la scène 3D, ainsi les sphères placées derrière semblent plus petites. En outre, la sphère support est représentée par un fil de fer transparent. Bien que les ombres soient couramment utilisées dans les scènes 3D, notre adaptation n'encourage pas l'utilisation d'ombres pour éviter de surcharger l'environnement. Ces recommandations constituent donc une scène de référence pour l'évaluation des techniques d'interaction pour les tâches de pointage en 3D; l'utilisation de cette scène de référence commune évite l'insertion de biais provenant d'une

mauvaise perception de la profondeur qui pourraient compromettre les résultats de l'évaluation de l'interaction dans les espaces 3D.

Ces travaux ont été réalisés dans le cadre de la thèse de Bénédicte Schmitt, doctorante de 2009 à 2014, que j'ai co-encadrée avec Emmanuel Dubois et Cédric Bach. Ces recherches sur l'adaptation de la norme et la plate-forme d'évaluation ont donné lieu à une publication dans la revue nationale TSI [SCHMITT et al., 2013] et à la conférence internationale SouthCHI [RAYNAL et al., 2013].

C 3 DIM – Modèle de description des dispositifs de pointage 3D

En plus du problème de l'évaluation de la tâche de pointage, nous nous sommes également intéressés à la phase de conception de techniques d'interaction utilisées dans un environnement virtuel 3D (EV 3D). Auparavant limités à une utilisation pour les jeux vidéo, ces environnements sont maintenant régulièrement utilisés dans divers contextes grand public tels que des expositions dans des musées [PECCHIOLI et al., 2011], ou même dans des magasins pour configurer l'intérieur de son habitat en fonction des meubles vendus [LEE et al., 2011]. Cependant, nous avons pu constater que les dispositifs proposés pour interagir avec ces différents EV 3D étaient souvent le traditionnel couple clavier / souris. Pourtant, les dispositifs d'interaction ont largement évolué ces deux dernières décennies et il existe maintenant de nouvelles formes avancées d'interaction, telles que l'informatique mobile [BERGÉ, SERRANO et al., 2014] et ambiante [ISHII et al., 1998], les interfaces spatiales [LAVIOLA et KEEFE, 2011] ou encore tangibles [WILLIAMS et al., 2011]. De plus, les formes avancées de techniques d'interaction offrent de nouveaux potentiels tels que le fait d'être basées sur des objets personnels (dispositifs ou artefacts), d'être intégrées dans l'environnement physique, d'être faciles à appréhender. Mais le développement d'applications interactives en 3D d'une part et les techniques d'interaction avancées d'autre part sont deux préoccupations qui sont le plus souvent considérées à travers des approches séparées.

A partir de ce constat, Il nous est apparu nécessaire de proposer un cadre de conception qui permette aux communautés IHM et 3D de se rapprocher en travaillant autour d'un modèle commun. Concrètement, notre cadre de conception est basé sur un modèle bien connu et fortement utilisé de chaque communauté (la tâche de l'utilisateur pour l'IHM et le graphe de scène pour la 3D). L'originalité de notre cadre se base sur l'identification des liens entre les tâches de l'utilisateur et les attributs impactés des éléments 3D représentés dans le graphe de la scène (cf. Figure 25).

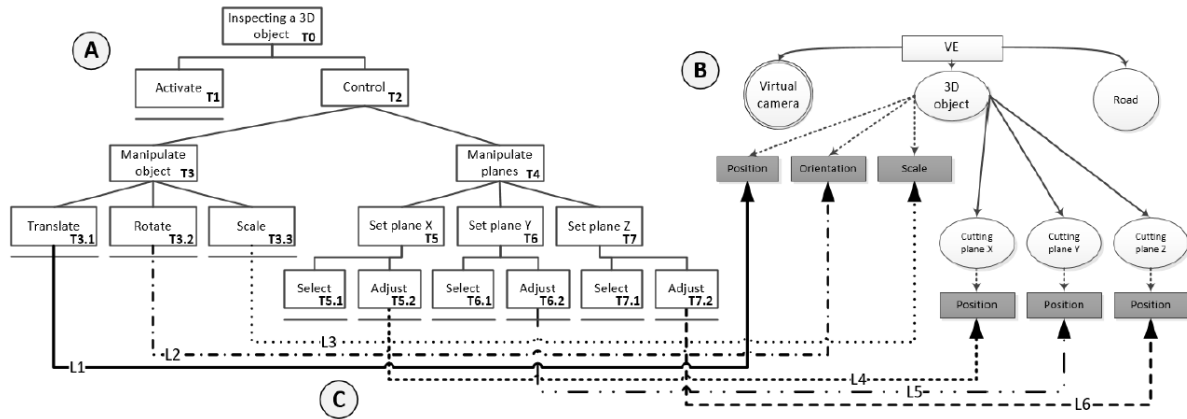


FIGURE 25 – Lien entre modèle de tâches et graphe de scène

Chaque lien créé entre une sous-tâche du modèle de tâche et un élément du graphe de scène nécessite l'utilisation d'une modalité d'interaction pour réaliser concrètement la sous-tâche et donc modifier le graphe de scène interactive décrivant la structure de l'environnement 3D. Pour caractériser toutes les étapes du lien pour passer de la tâche utilisateur à l'élément dans la scène 3D, nous avons élaboré la notation 3DIM (pour 3D Interaction Modality). Celle-ci est décomposée en 6 blocs (cf. Figure 26) qui représentent respectivement :

L'utilisateur, notamment en décrivant les parties (membres, articulations) impliquées, ses caractéristiques (par exemple, main dominante/non dominante), la posture du corps, mais aussi le niveau d'expertise de l'utilisateur final (expert, expérimenté ou débutant).

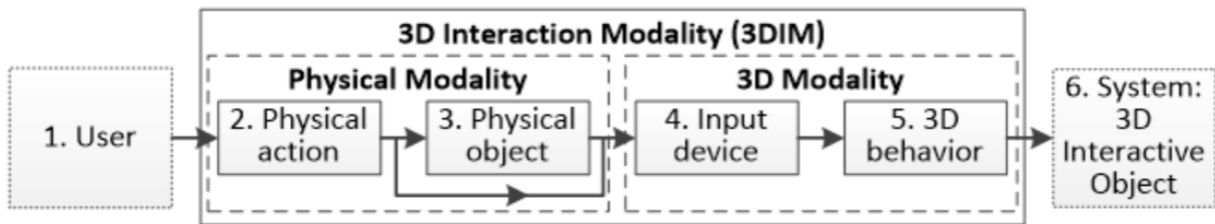


FIGURE 26 – Les 6 blocs d'un lien entre une sous-tâche et un élément du graphe de scène

L'action physique qui décrit les actions réalisées par l'utilisateur dans l'espace physique qui l'entoure. Les aspects les plus importants pour l'analyse d'une modalité d'interaction concernent la nature du mouvement effectué (discret ou continu), le type de mouvement (par exemple translation, orientation, geste), et le cadre de référence dans lequel ces gestes sont effectués (référence au monde, attaché à un objet physique statique ou mobile par exemple un écran, un smartphone ou une table).

L'objet physique décrit l'artefact physique potentiellement impliqué dans une modalité d'interaction 3D, ses caractéristiques qui impactent l'interaction (par exemple, la forme, la taille, la couleur, la texture, les degrés de liberté) et les contraintes physiques dues à d'autres objets physiques, à l'environnement physique, aux dispositifs utilisés ou aux membres de l'utilisateur.

Le dispositif d'entrée - ce bloc décrit le dispositif d'entrée utilisé pour capter les mouvements effectués par l'utilisateur. Il permet de définir le type d'informations recueillies (image, son, champ magnétique, position, force, angle, couple, état) et le nombre de degrés de liberté captés et utilisés. L'emplacement du dispositif d'entrée dans l'espace physique et les autres contraintes physiques ou spatiales (par exemple, sur la table, entre l'écran et l'utilisateur) y sont également détaillés.

Comportement 3D - ce bloc adopte un point de vue logiciel sur l'action réalisée dans l'environnement 3D. Il décrit, avec un haut niveau de description, la transformation logicielle des données entrantes en données sortantes. Pour décrire le comportement 3D complet d'une tâche, il est parfois nécessaire de combiner ou d'avoir plusieurs blocs de comportement 3D. Ce bloc donne une description textuelle du comportement et précise le rôle, l'objectif du comportement 3D qui devrait aider à identifier rapidement les différences ou les similitudes entre plusieurs blocs de comportements 3D.

L'objet interactif 3D qui détermine le type d'objet virtuel manipulé par la tâche (formes géométriques, maillage, nuage de points, etc.), ainsi que les attributs de l'objet qui sont affectés par le bloc de comportement 3D (position, rotation, échelle).

Pour aller plus loin dans la description des différents blocs, nous nous sommes appuyés sur des travaux de la littérature des communautés IHM et 3D abordant différents points de vue pour la conception de l'interaction avec un EV 3D. Nous avons ainsi établi pour chacun des 6 blocs présentés ci-dessus des liens avec les approches de conception existantes pour ce type de bloc. L'intérêt est que, dans des cas spécifiques, un concepteur peut vouloir décrire plus profondément l'un des six blocs : dans ce cas, la synthèse des approches que nous proposons constitue un support pour choisir un outil de conception complémentaire parmi les approches de conception appropriées déjà existantes. Ce cadre offre donc une vue globale sur l'interaction avec l'ensemble de l'EV 3D et non pas une vue limitée à une tâche atomique 3D spécifique et étroite (navigation, sélection ou manipulation) ou à une technologie spécifique. En outre, le cadre réconcilie les points de vue, les approches et les considérations de conception de l'IHM et de la 3D : les attributs influencés par ces deux communautés sont combinés dans le cadre. De plus, le cadre constitue un support pour fédérer et organiser les différentes approches en les positionnant selon leur impact sur les différentes facettes de l'interaction de l'utilisateur avec un 3DVE (les six blocs de 3DIM).

Enfin, pour faciliter l'analyse des techniques d'interaction conçues avec ce cadre, nous avons également proposé des propriétés quantitatives (par exemple le nombre de liens présents, le nombre d'objets physiques utilisés, le ratio de blocs partagés, etc.) et qualitatives qui permettent d'avoir des indicateurs sur la technique d'interaction proposée et ainsi la comparer plus facilement à d'autres techniques également modélisées dans notre cadre de conception.

Tous ces critères sont présentés de manière bien plus détaillée dans le mémoire de doctorat de Pierre-Louis Bergé [BERGÉ, 2015], doctorant que j'ai co-encadré avec Emmanuel Dubois et Minica Houry-Panchetti de 2011 à 2015.

D Synthèse

Dans ce chapitre, j'ai présenté les principaux travaux théoriques effectués sur la tâche de pointage et plus globalement la conception de techniques d'interaction dans des environnements virtuels 3D.

Ma première contribution a été une proposition d'alternative au modèle traditionnel de Fitts, reconnu et très fortement utilisé pour prédire le temps de pointage pour une tâche donnée. Nous avons décrit une nouvelle approche de l'apprentissage de la loi de Fitts au travers de la régression floue qui permet de représenter à la fois la tendance générale des données et la variation autour de cette tendance. Ceci nous permet, par rapport à la régression linéaire généralement utilisée, de distinguer les cas où la variation est faible (et donc la prédiction plus précise) des cas où elle est plus importante. L'ensemble flou décrivant les valeurs possibles du temps a d'autant plus de sens si cette variation est assez importante. Cette approche ouvre de nombreuses perspectives : elle pourrait servir pour une optimisation plus fine des interfaces proposant de nombreuses cibles (claviers logiciels, menus à plusieurs niveaux, etc.) et faire des analyses d'interface plus précises en utilisant, par exemple, les bornes de nos ensembles flous (pour étudier le pire et/ou le meilleur des cas d'utilisation d'une interface). Ce travail a été réalisé en collaboration avec Mathieu Serrurier, de l'équipe ADRIA de l'IRIT, et qui est spécialisé en Apprentissage. Notre travail à la frontière des deux disciplines a été présenté dans deux conférences de nos domaines respectifs ([RAYNAL et SERRURIER, 2007] et [SERRURIER et RAYNAL, 2008]).

Le reste de mes travaux théoriques sur le pointage s'est focalisé sur un contexte d'interaction plus précis : l'évaluation de la tâche de pointage dans un environnement virtuel 3D. S'il existe bien une norme pour mettre en œuvre l'évaluation de dispositif de pointage pour une interface traditionnelle en 2D, en revanche, il n'existait rien pour évaluer spécifiquement les dispositifs utilisés dans des environnements virtuels 3D. C'est pourquoi nous nous sommes attachés à proposer une adaptation de la norme existante pour la 2D au contexte plus spécifique de la 3D. Afin de faciliter son utilisation et essayer d'harmoniser les pratiques d'évalua-

tions, nous avons également proposé une plate-forme d'évaluation pour les tâches en 3D et qui respecte les recommandations proposées dans notre adaptation de la norme. Ce travail a été réalisé par Bénédicte Schmitt, doctorante que j'ai eu l'occasion de co-encadrer avec Emmanuel Dubois et Cédric Bach. Nos travaux sur l'évaluation de la tâche de pointage ont été publiés à IHM [SCHMITT, RAYNAL, DUBOIS et BACH, 2011] et 3DUI [SCHMITT et al., 2012] pour présenter l'adaptation de la norme et de manière plus complète avec la plate-forme d'évaluation dans la revue nationale TSI [SCHMITT et al., 2013] et à la conférence internationale SouthCHI [RAYNAL et al., 2013].

Enfin, notre dernière contribution est un cadre de conception pour aider à la conception de techniques d'interaction en EV 3D. Il est basé sur 1) la description des besoins des utilisateurs à travers un arbre de tâches, 2) l'identification et la caractérisation des attributs manipulés des objets 3D formant la EV 3D à travers le graphe de scène interactive, et 3) l'identification des liens entre les sous-tâches des utilisateurs et les attributs manipulés du graphe de scène interactive. Ces liens soutiennent l'interaction et peuvent être décrits plus précisément par la notation 3D Interaction Modality (3DIM). Cette notation fournit six concepts différents, appelés blocs 3DIM : utilisateur, action physique, objet physique, dispositif d'entrée, comportement 3D et objet interactif 3D. Chaque bloc contient un ensemble de caractéristiques avec différentes valeurs possibles afin d'affiner la description de la modalité 3DIM. Notre approche de conception est particulièrement originale, car elle offre une vue d'ensemble de la situation interactive qui regroupe les considérations de conception des communautés HCI et 3D. L'approche de conception proposée est également très utile, car le support d'analyse est intégré dans le cadre proposé et fait référence à des propriétés existantes et importantes identifiées en IHM et en 3D. En adoptant un point de vue qui rassemble des considérations traitées dans deux communautés différentes et en réutilisant explicitement des approches existantes, notre travail contribue à 1) obtenir une meilleure compréhension des éléments sous-jacents à une technique d'interaction 3D, 2) identifier de manière plus systématique, toutes les questions pertinentes pour la conception de nouvelles techniques d'interaction 3D et 3) faciliter le développement de techniques d'interaction 3D mieux adaptées aux non-experts. Ce travail a été réalisé par Louis-Pierre Bergé, doctorant que j'ai pu co-encadrer avec Emmanuel Dubois et Minica Houry-Panchetti. Nous avons présenté une partie de ce cadre à un workshop de la conférence EICS [BERGÉ, DUBOIS, HOURY-PANCHETTI et al., 2014].

Ces travaux théoriques sur le pointage m'ont également permis de participer à l'organisation d'ateliers sur l'analyse du mouvement [ADER et al., 2019; ADER et al., 2020; ADER et RAYNAL, 2020].

Chapitre 5

Conception de techniques d'interaction pour des environnements « complexes »

Comme nous avons pu le voir au travers des exemples présentés en introduction, les environnements numériques sont de plus en plus complexes : le nombre de données présentes dans ces environnements explose et amène les concepteurs à proposer des surfaces d'affichage de plus en plus grandes et des techniques de visualisation de données de plus en plus complexes. Mes travaux sur la conception de nouvelles techniques d'interaction ont donc principalement porté sur le pointage et la manipulation de données présentes sur de grandes surfaces d'affichages, ou dans des environnements plus complexes tels que les EV 3D.

Mes contributions peuvent se présenter sous 3 axes : 1) la conception de techniques d'interaction pour pointer des objets sur une grande surface d'affichage ; 2) la conception de techniques d'interaction basées sur le smartphone pour pointer et manipuler des objets d'un environnement virtuel 3D distant ou affiché sur le smartphone ; 3) la conception de techniques d'interaction multi degrés de liberté pour pointer et manipuler des éléments dans des environnements hétérogènes.

A Pointage sur une grande surface d'affichage distante

On parle de pointage indirect lorsque le dispositif de pointage est découplé de la surface d'affichage. Cette séparation donne lieu à deux espaces différents : d'une part, l'espace moteur (ou de contrôle), dans lequel l'utilisateur manipule le dispositif de pointage ; et d'autre part, l'espace d'affichage, dans lequel un pointeur est affiché et dont les mouvements sont contrôlés par l'utilisateur au moyen de son dispositif de pointage. Il existe principalement deux types de dispositifs de pointage : d'une part, les dispositifs isotoniques, comme la souris, où la force appliquée sur le dispositif est constante, et où le mouvement effectué avec le dispo-

sitif est pris en compte. D'autre part, les dispositifs isométriques, tels que les joysticks, où le dispositif reste à la même position, mais sur lequel l'utilisateur applique une force dans une direction. Un dispositif de pointage peut contrôler soit la position, soit la vélocité (vitesse) du pointeur. Le contrôle de la position est le plus souvent utilisé avec des dispositifs isotoniques, tandis que le contrôle de la vitesse est utilisé avec des dispositifs isométriques [ZHAI, 1995].

Les dispositifs les plus utilisés sont les dispositifs isotoniques. Cependant, lorsque l'espace moteur est beaucoup plus petit que l'espace d'affichage, il n'est pas possible de déplacer le pointeur d'un bout à l'autre de la surface d'affichage avec un seul déplacement du dispositif isotonique. Dans ce cas, l'utilisateur doit effectuer plusieurs mouvements pour déplacer le pointeur. Dans ce cas, il est nécessaire d'exécuter un mouvement pour ramener le dispositif de pointage à sa position initiale, sans que cela ne soit répercuté sur le pointeur. Ce mouvement est appelé « *clutching* » [HINCKLEY et al., 1994] et son exemple le plus connu est celui de la souris que l'on soulève pour la ramener à sa position d'origine sans que le pointeur ne bouge. Cependant, ce mécanisme présente l'inconvénient de ne pas être très efficace lorsque le déplacement final du pointeur nécessite plusieurs *clutching*. En effet, chaque phase de *clutching* nécessite un certain temps pendant lequel le pointeur ne bouge pas.

Notre premier axe de travail a été d'étudier différentes techniques d'interaction pour limiter la phase de *clutching*, en particulier pour des tâches de pointage sur de grandes surfaces d'affichage.

1 Manipulation d'un dispositif isométrique logiciel

Notre première étude a consisté à proposer un pointeur augmenté reprenant le principe du joystick physique : le Softstick. Celui-ci est représenté par un disque gris et d'une croix. Le centre de la croix sert de pointeur à l'écran. Ce disque contient un cercle bleu de plus petite taille. Le disque gris représente l'espace moteur dans lequel l'utilisateur peut bouger son dispositif et le cercle bleu représente la position du dispositif dans cet espace moteur. Le Softstick se déplace sur la surface d'affichage en fonction de la position du cercle bleu sur le disque : si le cercle est au centre du disque, le Softstick ne bouge pas. Dès que le cercle bleu n'est pas au centre du disque, le Softstick se déplace dans la direction et le sens donné par le vecteur formé par le centre du disque et celui du cercle bleu (cf. Figure 27 - gauche). Plus le cercle s'écarte du centre du disque et plus le Softstick se déplace rapidement.

Pour contrôler le déplacement du pointeur dans le Softstick, nous avons essayé deux techniques d'entrée basées sur les caractéristiques d'un UMPC (ancêtre du smartphone) : l'une, tangible, qui tire profit de la position de l'UMPC lui-même dans l'espace moteur ; et l'autre, tactile, qui se base sur la surface tactile de l'UMPC pour capturer la position d'un doigt ou d'un stylet.

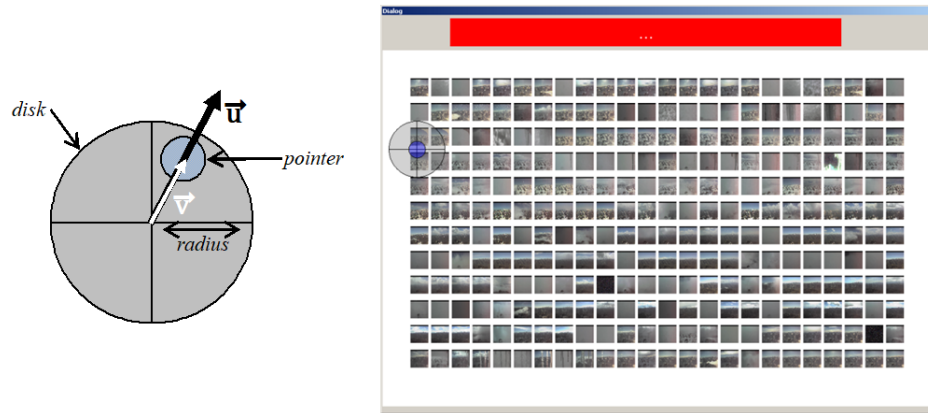


FIGURE 27 – Illustration du fonctionnement du Softstick (à gauche), et de son utilisation sur une interface (à droite)

Dans la première technique, la position du pointeur du Softstick dépend de la position de l'UMPC dans l'espace moteur. Par conséquent, les utilisateurs déplacent le pointeur par de grands mouvements du bras. Ces mouvements, peu rapides, limitent le besoin d'habileté et de concentration pour effectuer les déplacements du curseur. La seconde technique d'interaction repose sur l'écran tactile de l'UMPC. La position du pointeur du Softstick dépend ici de la position du stylet sur l'écran.

Nous avons comparé l'utilisabilité de ces deux techniques de pointage lors d'une étude avec des utilisateurs. Cette évaluation visait dans un premier temps à étudier l'applicabilité de la loi de Fitts dans ce contexte, avec une couche logicielle intermédiaire supplémentaire. Les données recueillies nous ont permis de conclure que le modèle prédictif de Fitts s'applique également à ce type de techniques d'interaction. D'autre part, l'analyse de ces données, notamment sur les performances de pointage a révélé que la technique tactile est plus efficace que la technique tangible pour les tâches de pointage dont l'indice de difficulté est inférieur à 4, tandis que la technique tangible est meilleure pour les tâches de pointage avec un indice de difficulté supérieur à 4.

Ce travail a été réalisé dans le cadre du doctorat de Bénédicte Schmitt et a été publié en 2010 lors de la conférence internationale NordiCHI [RAYNAL et al., 2010].

2 Techniques de pointage reposant sur l'élasticité du poignet

Les solutions tangibles, telle que celle présentée précédemment, peuvent engendrer des fatigues musculaires lors d'une utilisation longue. Afin de pallier cette limitation, nous nous sommes intéressés à des techniques de pointage tangibles où les mouvements de l'utilisateur sont plus réduits et provoquent moins de tensions musculaires.

Ce travail, réalisé par Théophanis Tsandilas, avait pour objectif de proposer des techniques d'interaction qui tirent profit des caractéristiques et contraintes biomécaniques du poignet. En effet, les mouvements du poignet sont limités par les articulations de la main. Les contraintes articulaires diminuent l'amplitude des mouvements, ce qui entrave les performances de l'utilisateur. C'est un problème majeur lorsque les utilisateurs doivent interagir sur de grandes surfaces d'affichage sans perdre en précision de pointage.

Pour ce travail, nous avons examiné trois techniques d'interaction uniquement basées sur les mouvements du poignet. L'utilisateur tient simplement dans sa main un petit dispositif de détection de mouvement (Ascension Flock of Birds 6-DOF) permettant de détecter les extensions et les flexions du poignet autour d'un axe vertical. La première technique, appelée « Pure Rate Control » considère le poignet comme un joystick naturel où la main au repos représente la zone neutre, puis plus l'angle entre l'avant-bras et la main est important, plus le pointeur se déplace rapidement.

La deuxième technique, nommée « Hybrid Control », fonctionne avec deux zones distinctes : la zone centrale autour de la position au repos du poignet est utilisée pour faire du pointage absolu par contrôle de position. Dans cette zone, la position du poignet est représentée par un pointeur que l'utilisateur peut déplacer en effectuant des mouvements du poignet sans avoir à effectuer de flexions trop importantes du poignet. Cette zone est représentée à l'écran par une fenêtre grise. Dès que le pointeur arrive en limite de cette fenêtre (et donc avec une flexion plus conséquente du poignet), la fenêtre se déplace à une vitesse donnée (cf. Figure 28). Cette technique reprend le fonctionnement des techniques dites « hybrides », comme par exemple celle proposée par Casiez et al. [CASIEZ et al., 2007], qui mêlent un fonctionnement par contrôle de vitesse (ici de la fenêtre) pour permettre de se déplacer plus rapidement d'un point à un autre et un contrôle de position pour affiner le pointage en fin de tâche (dans notre cas, contrôle du pointeur à l'intérieur de la fenêtre). Pour pointer un élément, l'utilisateur doit donc commencer par positionner la fenêtre sur l'élément souhaité, puis il termine son pointage avec le pointeur présent à l'intérieur de la fenêtre.

Enfin, la troisième technique, appelée « Non-Uniform Position Control », reprend le même principe que la technique précédente en proposant un pointage en deux étapes avec une fenêtre dans laquelle l'utilisateur manipule directement un pointeur, puis la possibilité de déplacer cette fenêtre en effectuant des mouvements plus amples du poignet. Dans cette technique, les deux types de déplacement (de la fenêtre et du pointeur) se font par un contrôle de position. En revanche, la fonction de transfert, qui permet de déterminer le déplacement du pointeur en fonction du mouvement effectué par le poignet, peut être différente pour les 2 zones. Ainsi, un gain faible à l'intérieur de la fenêtre permet d'avoir un déplacement moins rapide mais plus précis du pointeur. Et inversement, avec un gain fort, l'utilisateur pourra déplacer rapidement la fenêtre mais avec moins de précision. En jouant sur les fonctions de

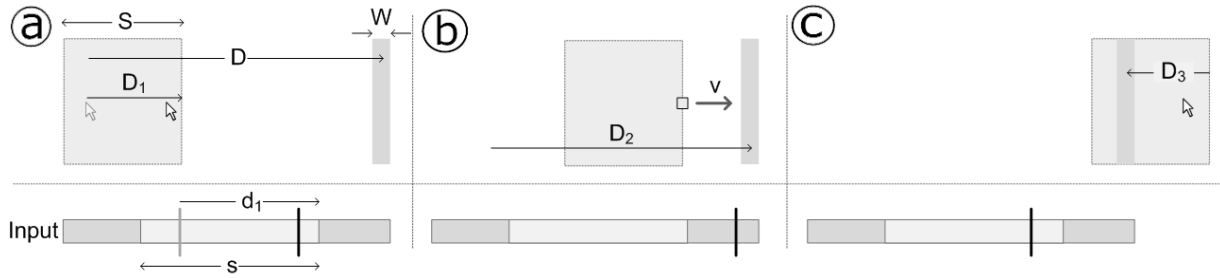


FIGURE 28 – Étapes d’une tâche de pointage avec la technique hybride lorsque $D > D_1$. (a) Le curseur se déplace sur une distance D_1 , qui correspond à un mouvement d_1 le long de la plage isotonique de l’entrée. (b) La fenêtre se déplace vers la cible, qui apparaît à une distance D_2 du centre initial de la fenêtre. La vitesse v de la fenêtre augmente au fur et à mesure que la position de l’entrée s’étend vers ses côtés extrêmes. (c) Le curseur se déplace vers la cible en couvrant une distance D_3 .

transfert des 2 zones, nous pouvons alors trouver un bon compromis entre se déplacer rapidement sur de longues distances puis terminer le pointage plus précisément dans la fenêtre.

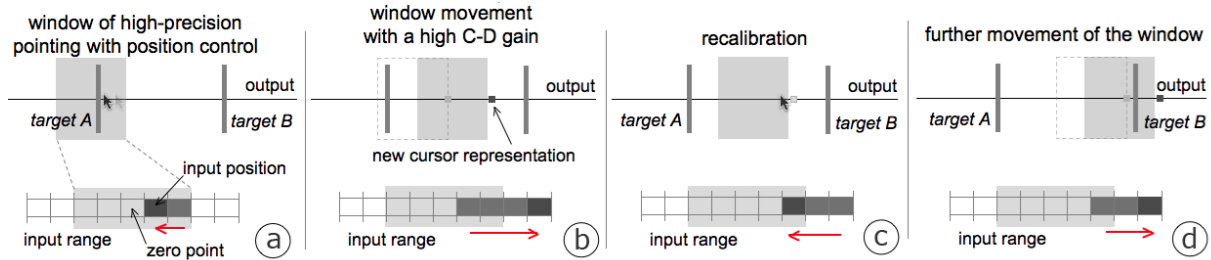


FIGURE 29 – Contrôle de position non-uniforme. (a) Pointage absolu dans la zone centrale de la plage d’entrée avec un faible gain C-D. (b-d) Recalibrage de la position avec des mouvements de va-et-vient

Nos résultats expérimentaux montrent que ces deux techniques sont plus précises que le contrôle de vitesse pur et leurs performances s’améliorent lorsque la précision de l’entrée augmente. La technique hybride s’est avérée particulièrement efficace pour équilibrer la précision et la vitesse dans la plage de 61 à 241 unités d’entrée. Elle a surpassé la technique de contrôle de position non-uniforme dans les longues distances cibles de plus de 1000 pixels. Néanmoins, notre analyse suggère que les performances de la technique de contrôle de position sont plus rapides lorsque la précision des entrées s’améliore.

Les résultats préliminaires de ce travail ont été publiés sous forme de travaux en cours à la conférence CHI en 2010 [TSANDILAS et al., 2010], puis les résultats détaillés de l’évaluation de ces 3 techniques ont ensuite été publiés à la conférence internationale Interact en 2013 [TSANDILAS et al., 2013].

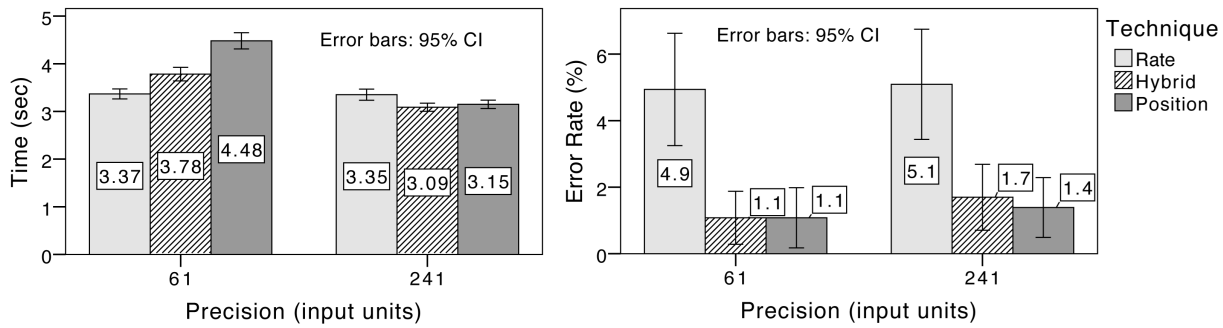


FIGURE 30 – Résultats

B Interaction avec un environnement virtuel 3D

1 Environnement virtuel 3D affiché sur une surface distante

Comme nous l'avons vu précédemment, l'utilisation de contenus 3D dans un environnement interactif devient de plus en plus fréquente : on trouve maintenant les EV 3D dans de nombreuses activités privées [MAGLIOCCETTI et al., 2012; VAJK et al., 2008], professionnelles [SONG et al., 2011; JEON et al., 2010] ou publiques [MARTON et al., 2014].

L'utilisation d'un smartphone pour soutenir l'interaction avec la EV 3D dans ces situations de la vie quotidienne présente de nombreux avantages potentiels : fournir une commande d'interaction à distance multi-capteurs évitant l'occlusion ; exploiter un dispositif personnel et familier ; afficher un retour supplémentaire sur l'écran du smartphone.

L'utilisation du smartphone pour interagir avec du contenu 3D sur un écran distant a déjà été envisagée, par exemple pour manipuler un plan de coupe [SONG et al., 2011], pour naviguer, sélectionner ou manipuler un objet 3D [KRATZ et al., 2012]. Les combinaisons de mouvements de doigts [TELKENAROGLU et CAPIN, 2013], d'orientations de l'appareil [LEE et al., 2011] et de gestes autour de l'appareil [KRATZ et al., 2012] représentent de nombreuses alternatives pour mettre en œuvre des techniques d'interaction 3D basées sur les smartphones. Cependant, parmi ces travaux, moins d'attention a été accordée à l'utilisation combinée de techniques autour de l'appareil, basées sur un smartphone, avec une scène 3D affichée sur un écran distant. Pourtant, les techniques d'interaction « autour du smartphone » ont déjà été explorées avec succès dans d'autres contextes [JONES et al., 2012; KETABDAR et al., 2010].

L'objectif du travail réalisé dans le cadre du doctorat de Louis-Pierre Bergé était de concevoir une nouvelle technique d'interaction impliquant des gestes autour d'un smartphone pour manipuler un objet 3D affiché sur un écran distant.

Dans cette technique, la main non dominante est utilisée pour tenir le smartphone. Le plan défini par le smartphone est mis en correspondance avec le plan (X, Z) de la scène 3D, c'est-

à-dire le sol de l'environnement virtuel, alors que l'axe Y de la scène 3D est associé à la perpendiculaire du smartphone. Le centre du smartphone représente l'origine du cadre de référence. Les mouvements de la main dominante sont donc évalués par rapport au Smartphone. Dans notre technique, les mouvements de translation en l'air sont mesurés autour du smartphone et sont appliqués à l'objet 3D manipulé. La rotation des objets repose sur le déplacement de la main dominante dans un mouvement circulaire au-dessus du smartphone, ce qui déclenche une rotation de l'objet manipulé dans l'axe Z. Plus généralement, tout mouvement circulaire autour du smartphone va déclencher une rotation des objets manipulés en 3D selon une combinaison des 3 axes.

L'intérêt de cette technique est qu'elle permet à l'utilisateur de choisir l'orientation du smartphone à sa convenance et donc le cadre de référence des mouvements de pointage qu'il effectue. Il peut adopter une orientation confortable du smartphone devant son buste et ensuite interagir confortablement et efficacement avec la main dominante autour du smartphone. Un autre avantage de cette technique est que le feedback affiché sur l'écran du smartphone est toujours visible par l'utilisateur.

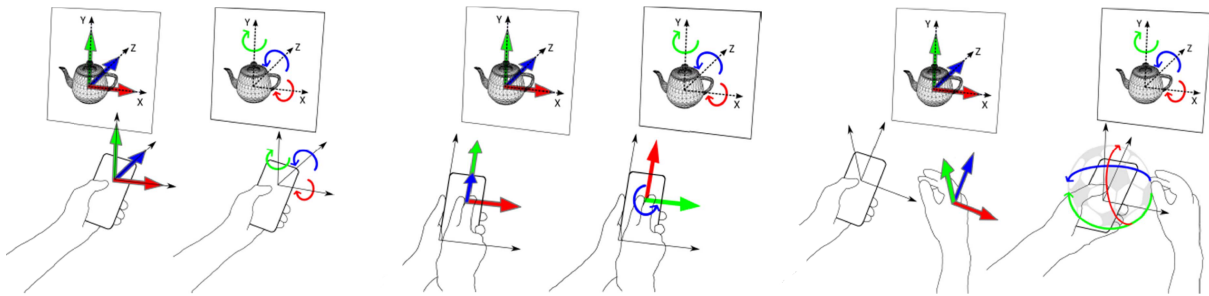


FIGURE 31 – Les 3 techniques

Nous avons comparé cette technique d'interaction avec une technique tangible « Avec le Smartphone » et une technique tactile « Sur le Smartphone ». Dans le cas de la technique tangible, le corps de l'utilisateur sert de référence. Le plan horizontal perpendiculaire à l'utilisateur debout est mis en correspondance avec le plan (X, Z) de la scène 3D, et l'axe Y de la scène 3D est donc parallèle à l'utilisateur debout. Les mouvements de translation du smartphone sont calculés par rapport au corps de l'utilisateur et appliqués à l'objet 3D manipulé. Les rotations de l'objet dépendent des rotations du smartphone.

Enfin, la technique tactile est une implémentation de la technique Dual Finger [TELKENAROGLU et CAPIN, 2013]. Les translations de l'objet le long de l'axe Y sont réalisées en déplaçant deux doigts conjointement vers le haut ou le bas de l'écran. Si les deux doigts sont écartés, le plan défini par l'écran du smartphone est mis en correspondance avec le plan (X, Z) de la scène 3D. La rotation repose sur le mouvement de trois doigts différents pour contrôler les trois degrés de liberté.

Cette technique Dual Finger nous a servi de technique de référence pour l'évaluation de nos techniques d'interaction. En effet, les résultats obtenus dans [TELKENAROGLU et CAPIN, 2013] montraient de meilleures performances avec Dual Finger par rapport à d'autres techniques d'interaction tactile connues (la technique Z [MARTINET et al., 2010] pour la translation et la technique Arc-Ball [BOWMAN et al., 2004] pour la rotation).

Notre expérience contrôlée montre que les techniques « Autour » et « Avec » sont plus performantes que la technique « Sur » dans les tâches de translation et de rotation. Bien que les techniques « Autour » et « Avec » donnent des résultats similaires dans les tâches de translation 3D, un compromis apparaît entre ces deux techniques : la technique « Avec » donne de meilleurs résultats que la technique « Autour » lors de la rotation d'un objet 3D, mais la technique « Autour » est la plus attrayante et la préférée.

2 Environnement 3D affiché sur Smartphone

La majorité des travaux concernant le pointage dans les EV 3D se sont focalisés sur un affichage de ce dernier soit sur des surfaces d'affichage traditionnelles (écrans d'ordinateur, ou projections sur grandes surfaces horizontales ou verticales, par exemple); soit dans des environnements plus immersifs et appropriés à de la 3D [GROSSMAN et BALAKRISHNAN, 2006]. L'affichage et l'interaction avec un EV 3D sur Smartphone est plus difficile car la surface d'affichage proposée par l'écran du dispositif est beaucoup plus petite. De ce fait, deux solutions sont possibles pour afficher des EV 3D complexes sur ces petits dispositifs. La première consiste à n'afficher qu'une partie de l'EV 3D pour conserver une taille d'objets assez grosse. Cependant, si l'EV 3D n'est pas affiché entièrement, l'utilisateur a une moins bonne perception de la scène dans son ensemble. De plus, l'objet à atteindre n'est pas forcément visible à l'écran et cela nécessite donc une phase de navigation préalable au pointage pour rechercher l'objet désiré.

L'autre solution consiste à afficher l'ensemble de l'EV 3D. Chaque objet a alors une taille plus petite. De plus, la petite taille de ces écrans rend la manipulation d'un pointeur difficile notamment pour pointer précisément un objet. Une technique couramment utilisée consiste à utiliser un volume pour sélectionner plusieurs objets [DANG et al., 2003; LIANG et GREEN, 1993], puis sélectionner l'objet parmi ceux pré-sélectionnés dans le volume. Cependant, les techniques de sélection proposées ne semblent pas adaptées à une utilisation sur Smartphone. L'utilisation d'un menu pour une sélection parmi plusieurs objets risquerait de masquer l'ensemble de l'EV 3D.

C'est pourquoi, nous nous sommes intéressés à la sélection d'un élément dans un ensemble d'objets pré-sélectionnés. Pour éviter les problèmes de précision du pointage avec un pointeur et d'occlusion avec l'affichage d'une liste ou d'un menu, nous avons choisi de nous

focaliser sur la solution consistant à passer le focus de sélection d'un objet à un autre. La technique Circulation [DANG et al., 2003] permet de passer d'un objet à un autre au moyen d'un appui sur un bouton du dispositif de pointage. Le premier objet sélectionné dans le sous-ensemble déjà pré-sélectionné est celui se trouvant le moins éloigné du point de vue de l'utilisateur en profondeur. A chaque appui sur le bouton, le système sélectionne l'objet suivant dans la profondeur. Par conséquent, le temps de sélection peut rapidement augmenter si l'objet à sélectionner est situé après un nombre important d'objets dans la profondeur, car l'utilisateur devra passer un à un tous les objets précédant l'objet souhaité.

Pour répondre à ce problème de sélection dans un grand ensemble d'éléments, nous avons proposé la technique DichotoZoom qui reprend le principe de la dichotomie. Le but est de diviser par deux le nombre de cibles pré-sélectionnées à chaque fois que l'utilisateur effectue une action de sélection. A chaque étape, l'ensemble des cibles pré-sélectionnées est divisé en deux ensembles de cibles de taille égale. Les ensembles sont constitués en fonction de la profondeur de chaque cible par rapport au point de vue de l'utilisateur. Les $N/2$ cibles les plus proches du point de vue de l'utilisateur constituent un premier ensemble, tandis que les $N/2$ plus éloignées composent le second ensemble. L'utilisateur choisit alors l'ensemble dans lequel se situe la cible voulue. L'EV 3D n'affiche plus que le sous-ensemble des cibles choisi et réajuste le point de vue de l'utilisateur si c'est le sous ensemble le plus éloigné qui a été choisi. L'opération est alors renouvelée sur ce nouveau sous-ensemble jusqu'à ce que l'objet souhaité se trouve au premier plan de l'EV 3D et que l'utilisateur puisse le sélectionner sans contraintes (cf. Figure 32). Une première évaluation comparative des performances de pointage de notre technique DichotoZoom par rapport à la technique de référence Circulation nous a permis de montrer que la technique DichotoZoom permet de sélectionner rapidement un objet à l'intérieur de cet ensemble. De plus, quel que soit le nombre d'éléments, le nombre d'actions à réaliser reste inférieur à celui observé avec Circulation.

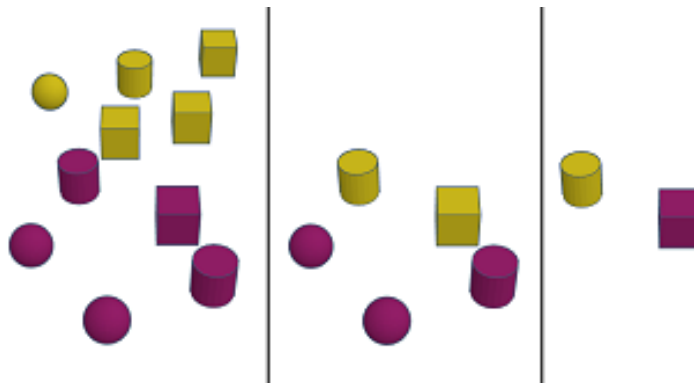


FIGURE 32 – Exemple de sélection dans un ensemble d'éléments pré-sélectionnés : la cible à atteindre est le cube représenté en violet sur l'image de gauche

D'autre part, la technique DichotoZoom fonctionne de manière discrète et ne nécessite finalement que deux actions distinctes de la part de l'utilisateur pour pouvoir sélectionner la

zone adéquate : sélection de la zone proche ou de la zone éloignée du point de vue de l'utilisateur.

Pour affiner la technique d'interaction, nous avons étudié l'utilisation de différentes modalités d'interaction contribuant à une distinction rapide entre les deux zones. Nous avons choisi d'étudier la sélection par actions discrètes (e.g. appuyer sur un bouton) et par action continue (e.g. un geste distinctif), et ce pour deux cas : dans une forme physique (utilisation du Smartphone lui-même) ou numérique (interaction directe sur l'écran tactile du Smartphone). Ainsi, la combinaison de ces deux caractéristiques nous a amené à étudier 4 modalités.

Une seconde évaluation comparant les 4 techniques d'interaction proposées a montré que les techniques discrètes (physiques ou logiciels) étaient les plus efficaces (en vitesse et en précision) et les préférées des utilisateurs pour utiliser la technique DichotoZoom.

Cette étude a été menée par Eliane Balaa durant son doctorat que j'ai co-encadré avec Emmanuel Dubois. Les résultats détaillés de ces études ont été publiés à la conférence IHM en 2018 [BALAA et al., 2018].

C Pointage dans un environnement hétérogène

Un élément essentiel lors de la conception d'un dispositif d'entrée est le nombre de degrés de liberté qu'il offre à l'utilisateur pour interagir avec l'environnement numérique. Les dispositifs peuvent offrir jusqu'à 6 degrés de liberté avec les 3 translations et les 3 rotations possibles dans un référentiel en 3 dimensions. Plus un dispositif a de degrés de liberté, et plus il permettra à l'utilisateur de réaliser des actions différentes sans procéder à des combinaisons complexes. Les éléments d'un environnement virtuel 3D pouvant souvent être manipulés selon 3 axes en translation et 3 axes en rotation, il est préférable de proposer des dispositifs d'entrée avec un maximum de degrés de liberté. Les techniques « Autour » et « Avec » le smartphone présentées à la section précédente avaient l'avantage de proposer 6 degrés de liberté. Cependant, ces solutions sont conçues pour une utilisation en mobilité, debout, et pour des périodes d'interaction assez courtes. Des interactions avec ce type de dispositif pourraient s'avérer fatigantes lors d'une utilisation de longue durée.

A l'inverse, les dispositifs d'entrée pour situations de travail de bureau ont souvent beaucoup moins de degrés de liberté. Par exemple, la souris traditionnellement utilisée n'a que 2 degrés de liberté. Or, comme nous l'avons vu, les environnements virtuels 3D sont aussi utilisés dans des contextes professionnels, dans des applications utilisées sur des ordinateurs de bureau. C'est pourquoi, nous nous sommes intéressés à la conception d'un dispositif d'entrée

utilisable sur un plan de travail, tout en augmentant son nombre de degrés de liberté. Nous avons ainsi proposé le dispositif Roly-Poly Mouse.



FIGURE 33 – Exemple de sélection dans un ensemble d'éléments pré-sélectionnés : la cible à atteindre est le cube représenté en violet sur l'image de gauche

Ce facteur de forme présente plusieurs avantages clés qui le rendent intéressant pour la construction d'un dispositif d'entrée. Comme le dispositif se déplace librement dans le plan, il peut être utilisé comme un dispositif isotonique tel que la souris. En même temps, le dispositif roule et s'auto-repositionne lorsqu'il est relâché, comme un dispositif isométrique. Le fond hémisphérique offre une grande surface incurvée permettant une large gamme de valeurs d'entrée possibles. Le dispositif est symétrique le long de son axe de rotation (z) et peut donc être utilisé dans n'importe quelle orientation soit en l'inclinant soit en effectuant une rotation du dispositif sur lui-même. Les trois gestes simples (translation, inclinaison et rotation) peuvent également être combinés pour créer des gestes composés. Pour aboutir à une solution exploitable, nous avons réalisé plusieurs prototypes pour trouver la meilleure forme du dispositif et la manière de capturer les mouvements effectués avec la Roly-Poly Mouse. De plus, nous avons également effectué plusieurs études préliminaires pour analyser les amplitudes maximums des rotations et des inclinaisons et leur séparabilité [PERELMAN et al., 2014].

Le dispositif Roly-Poly Mouse, avec ses 5 degrés de liberté et les combinaisons possibles entre les différents gestes nous a permis d'étudier différents cas d'utilisation possibles. Le premier a été d'évaluer la pertinence de notre dispositif pour pointer et manipuler des objets dans un environnement virtuel 3D. Nous avons ainsi mené une évaluation comparative entre notre dispositif et la souris SpaceMouse, régulièrement utilisée comme dispositif pour de la manipulation d'objets 3D. Les résultats ont révélé que la RPM est 31% plus rapide que la SpaceMouse pour effectuer les translations, et équivalente pour les rotations. D'autre part, l'évaluation qualitative a montré que les utilisateurs ont préféré notre dispositif hémisphérique pour l'interaction 3D.

Au-delà de la possibilité d'être utilisé comme dispositif d'entrée pour interagir avec un EV 3D, nous avons également étudié l'utilisation de Roly-Poly Mouse pour contrôler à distance des menus. En effet, dans plusieurs applications, les menus avec les différentes fonctionnalités sont éloignés de l'espace de travail. L'utilisateur doit alors déplacer le pointeur de son dispositif de pointage pour aller sélectionner l'item souhaité. Par conséquent, l'utilisateur

effectue de nombreux aller/retours entre l'espace de travail et la barre de menu, ce qui est préjudiciable pour son activité. L'autre solution consiste à utiliser les raccourcis clavier pour activer les fonctionnalités souhaitées. Mais cela nécessite d'une part de connaître les raccourcis clavier associés aux fonctionnalités, et d'autre part, cela oblige l'utilisateur à manipuler 2 dispositifs : le clavier et la souris. Pour pallier ce problème, nous avons exploré l'utilisation de Roly-Poly Mouse à la fois comme dispositif de pointage dans l'espace de travail (à la manière d'une souris traditionnelle faisant des translations), mais aussi en tant que dispositif pour sélectionner les items dans les différents menus en utilisant les inclinaisons et rotations de Roly-Poly Mouse. Nous avons appelé cette adaptation Rolling Menu.

Nous avons confronté Rolling-Menu à différents types de barres d'outils qui correspondent à des usages concrets dans des applications courantes : des barres d'outils simples comportant jusqu'à 14 éléments sont par exemple utilisées dans Thunderbird (réception de courrier, écriture, adresse, etc.); les barres d'outils déroulantes contenant jusqu'à 8x8 éléments sont par exemple utilisées dans Gimp; une barre d'outils de type grille comportant jusqu'à 9x9 éléments correspond à la palette d'outils dans Photoshop. Le Rolling-Menu peut donc facilement être intégré dans des applications quotidiennes.

Sur la base de différentes dimensions de conception, nous avons proposé 8 versions de Rolling-Menu que nous avons comparées à la souris pour sélectionner un élément de niveau supérieur dans une barre d'outils simple, ou des sous-éléments dans des barres d'outils plus complexes, c'est-à-dire une barre d'outils déroulante et une grille. Nos études auprès des utilisateurs ont démontré que deux menus déroulants réduisent le temps de sélection d'un élément ou d'un sous-élément de niveau supérieur, tout en conservant une bonne ou très bonne précision.

D Synthèse

Ce chapitre présente mes principales contributions concernant les prototypes conçus et évalués pour répondre principalement à la problématique du pointage dans des environnements « complexes ». Trois types d'environnements ont été ciblés : les grandes surfaces d'affichage sur lesquelles la distance à parcourir pour atteindre une cible peut être très grande; les environnements virtuels 3D qui nécessitent des dispositifs avec plus de degrés de liberté pour pouvoir pointer, puis manipuler correctement les objets de la scène; enfin les environnements où les données peuvent être représentées de manière hétérogènes et réparties à différents endroits de l'espace numérique.

Pour répondre à la problématique du pointage sur grandes surfaces d'affichage, nous nous sommes intéressés dans un premier temps à une solution logicielle en proposant le SoftStick, composant logiciel qui reprend le principe de fonctionnement du joystick. Ce dispositif logi-

ciel a l'avantage de pouvoir être utilisé avec tout type de dispositif de pointage, et nous avons montré qu'il conservait les mêmes propriétés qu'un dispositif de pointage physique notamment en respectant le modèle de la loi de Fitts. Ce travail a été réalisé avec Bénédicte Schmitt, doctorante de 2009 à 2014, que j'ai co-encadrée avec Emmanuel Dubois et Cédric Bach. Le résultat de ce travail a été publié et présenté en 2010 lors de la conférence internationale NordiCHI [[RAYNAL et al., 2010](#)].

Notre seconde contribution dans ce domaine a été d'étudier les caractéristiques et contraintes du poignet pour utiliser au mieux toutes les possibilités liées à son amplitude de mouvement et son élasticité. Nous avons proposé 3 techniques d'interaction uniquement basées sur les mouvements du poignet pour déplacer un pointeur dans une grande surface d'affichage. Ce travail a été effectué avec Theophanis Tsandilas, post doctorant de 2009 à 2010 que j'ai co-encadré avec Emmanuel Dubois. Les résultats des études préliminaires à ce travail avaient été publiés et présentés sous forme de travaux en cours à la conférence CHI en 2010 [[TSANDILAS et al., 2010](#)], puis les résultats détaillés de l'évaluation de ces 3 techniques ont ensuite été publiés et présentés à la conférence internationale Interact en 2013 [[TSANDILAS et al., 2013](#)].

Concernant le pointage et la manipulation d'objets dans un environnement virtuel 3D, nous nous sommes plus particulièrement intéressés à l'utilisation du smartphone comme dispositif d'entrée. Notre première contribution a été d'étudier l'utilisation du smartphone pour une interaction avec un environnement virtuel 3D affiché sur une surface distante. Dans ce contexte, nous nous sommes intéressés aux interactions qui pouvaient être proposées « sur », « avec » et « autour » le smartphone. Ce travail a été réalisé par Louis-Pierre Bergé, doctorant de 2011 à 2015, que j'ai co-encadré avec Emmanuel Dubois et Minica Houry-Panchetti. Les résultats de ces travaux sur le smartphone ont été publiés et présentés à la conférence internationale SUI en 2015 [[BERGÉ et al., 2015](#)].

Nous avons également étudié les interactions avec un smartphone dans le cas où l'environnement virtuel 3D est directement affiché sur le Smartphone. Les contraintes sont plus fortes que sur un écran distant car la surface d'affichage y est beaucoup plus réduite, et le smartphone ne peut pas être manipulé aussi librement car l'utilisateur doit toujours avoir une vue sur l'EV 3D. Dans ce contexte, nous avons étudié 4 modalités d'interaction différentes pour sélectionner un objet dans un sous ensemble d'objets présélectionnés. Ce travail a été effectué par Eliane Balaa, doctorante de 2011 à 2018 que j'ai co-encadrée avec Emmanuel Dubois. Les 4 techniques proposées et les résultats obtenus ont été publiés et présentés à la conférence nationale IHM en 2018 [[BALAA et al., 2018](#)].

Enfin, notre dernier axe de recherche a été la conception et l'évaluation d'un dispositif d'entrée à plusieurs degrés de liberté pour faciliter les interactions dans des situations d'interaction plus complexes. Ce travail a été réalisé en étroite collaboration avec Emmanuel Dubois et Marcos Serrano. Ces travaux ont abouti à plusieurs publications nationales et internatio-

nales pour présenter les problématiques de conception [PERELMAN et al., 2014; PERELMAN et al., 2015a; PERELMAN et al., 2016] et son évaluation dans différents cas d’usage [DUBOIS et al., 2017; DUBOIS et al., 2018].

Troisième partie

Synthèse et perspectives

Synthèse

Ce mémoire résume les travaux que j’ai pu mener depuis que j’ai été recruté à l’université Paul Sabatier en tant que maître de conférences en 2006. Ces travaux se sont concentrés principalement autour de deux grandes thématiques : d’une part la saisie de texte, et d’autre part, le pointage et la manipulation d’éléments dans des environnements numériques complexes. Pour aborder ces deux thématiques, mes contributions ont été de deux types : d’une part des travaux théoriques afin de mieux comprendre et structurer les enjeux scientifiques ; et d’autre part, des contributions plus pratiques avec la conception et l’évaluation de plusieurs prototypes répondant à une problématique donnée.

Mes contributions théoriques sur la saisie de texte ont porté sur la proposition d’un cadre de description permettant de décrire tout type de clavier logiciel, ainsi que sur l’étude de l’utilisation d’une liste de mots, et notamment sur la modélisation des performances théoriques que pourrait obtenir un utilisateur avec ce type de système de saisie de texte.

D’autre part, j’ai travaillé à la conception, au développement et à l’évaluation de plusieurs prototypes de systèmes de saisie de texte avec comme enjeu scientifique d’améliorer l’intégration des résultats proposés par les systèmes de prédiction de mots (WordTree, CentraList) ou de caractères (Floodkey [AULAGNER et al., 2010], SpreadKey [MERLIN et RAYNAL, 2009], Semantic keyboard [RAYNAL et al., 2014], SlideKey [RAYNAL et MARTIN, 2020]), ou de proposer un système de saisie de texte sur smartphone utilisable par les personnes déficientes visuelles (Duck). Nous nous sommes généralement appuyés sur une démarche de conception centrée utilisateur pour la conception de ces différents prototypes [TRUILLET et RAYNAL, 2013, TRUILLET et al., 2019]. Nous avons également conçu une architecture logicielle et instancié ces principales briques logicielles pour concevoir et développer la majorité des prototypes que nous avons proposés, ainsi qu’une majorité des systèmes de saisie de texte logiciels existants. Enfin, nous avons également proposé un outil permettant de modéliser la disposition des touches et liste de mots et simuler les performances théoriques des dispositions modélisées.

Sur cette thématique, j'ai eu l'opportunité d'encadrer 3 doctorants : Georges BADR (2007 à 2011), Bruno MERLIN (2008 à 2011), et Philippe ROUSSILLE (2013 à 2017). En outre, j'ai également encadré 8 étudiants de Master ou équivalent qui ont réalisé leur stage de fin d'études sur la thématique de la saisie de texte. L'ensemble de mes travaux sur la saisie de texte a donné lieu à la publication d'un chapitre de livre, d'une revue internationale, d'une revue nationale, de 19 articles dans des conférences ou workshops internationaux avec comité de lecture et de 5 articles dans des conférences ou workshops nationaux avec comité de lecture.

Mes travaux théoriques sur le pointage et la manipulation d'objets ont porté sur 3 contributions principales, respectivement autour de la modélisation de la loi de Fitts, de l'adaptation de la norme ISO 9241-9 relative à l'évaluation des dispositifs de pointage, et enfin sur la définition d'un cadre de conception pour les dispositifs d'entrée utilisés dans un environnement virtuel 3D.

Plusieurs prototypes ont également été conçus, développés et évalués pour répondre aux problématiques du pointage sur de grandes surfaces d'affichages (SoftStick, et des techniques d'interaction basées sur les seuls mouvements du poignet), de la manipulation d'objet dans des environnements virtuels 3D avec un smartphone (technique « autour » du smartphone, Dichotozoom) et enfin la réalisation de dispositif avec de multiples degrés de liberté pour accroître les possibilité d'interaction des utilisateurs (Roly-Poly Mouse).

Pour réaliser ces travaux sur le pointage, j'ai co-encadré 1 post-doctorant (Theophanis TSANDILAS de 2009 à 2010) et 3 doctorants, tous en collaboration et sous la direction d'Emmanuel DUBOIS : Bénédicte SCHMITT (2009 à 2014), Louis-Pierre BERGE (2011 à 2015) et Eliane BALAA (2011 à 2018). 4 étudiants de Master ou équivalent ont également réalisé leur stage de fin d'études sur la thématique du pointage. L'ensemble de mes travaux sur cette thématique a donné lieu à la publication d'une revue internationale, d'une revue nationale, et de 13 articles dans des conférences ou workshops internationaux avec comité de lecture et 7 articles dans des conférences ou workshops nationaux avec comité de lecture.

Le Tableau 3 synthétise et regroupe l'ensemble de mes contributions selon la thématique et le type de contribution. A côté de chacune des contributions, sont ajoutées entre parenthèses les initiales du doctorant ou de la doctorante que j'ai (co-)encadré(e) et qui a contribué à ce travail. Enfin, les références vers les publications associées au travail sont données entre crochets.

Au delà des travaux présentés dans ce mémoire, j'ai également travaillé sur d'autres projets. Depuis mes débuts en recherche, je me suis souvent impliqué dans le domaine des technologies d'assistance pour faciliter l'autonomie des personnes ayant une déficience visuelle. Après avoir travaillé sur l'accessibilité du web [RAYNAL et SERRURIER, 2002; MAUREL et al., 2003; MAUREL et al., s. d.], j'ai participé au projet NAVIG qui avait pour enjeu d'aider les

TABLEAU 3 – Synthèse de mes contributions

	Saisie de texte	Pointage et manipulation d'éléments
Contributions théoriques	- Espace de description DES-TIT - Architecture logicielle et KeySpec (BM) [MERLIN et al., 2012a] - Suivi du regard sur les listes de prédiction (GB) - Impact des connaissances utilisateur sur la saisie (BM) [MERLIN et RAYNAL, 2012]	- Fuzzy Fitts [RAYNAL et SERRURIER, 2007; SERRURIER et RAYNAL, 2008] - Adaptation de la norme ISO 9241-9 (BS) [SCHMITT, RAYNAL, DUBOIS et BACH, 2011; SCHMITT et al., 2013; RAYNAL et al., 2013] - 3DIM (PLB) [BERGÉ, DUBOIS, HOURS-PANCHETTI et al., 2014] - DECO [PERELMAN et al., 2016] - Classification des TI pour EV3D sur smartphone (EB) [BALAA et al., 2014]
Contributions pratiques (Techniques d'interaction)	- WordTree (GB) [BADR et RAYNAL, 2009b; BADR et RAYNAL, 2009c; BADR et RAYNAL, 2010] - CentralList (GB) [BADR et RAYNAL, 2011] - FishEye Keyboard [RAYNAL et al., 2007] - FloodKey [AULAGNER et al., 2010] - Spreadkey (BM) [MERLIN et RAYNAL, 2009; MERLIN et RAYNAL, 2010] - Semantic Keyboard (BM) [RAYNAL et al., 2014] - Duck (PR) [ROUSSILLE et al., 2013; ROUSSILLE et al., 2015; RAYNAL et ROUSSILLE, 2017; RAYNAL et ROUSSILLE, 2018]	- SoftStick (BS) [RAYNAL et al., 2010] - Interaction poignet (TT) [TSANDILAS et al., 2010; TSANDILAS et al., 2013] - Smartphone-based 3D Navigation technique (PLB) [BERGÉ, PERELMAN, RAYNAL et al., 2014; BERGÉ, PERELMAN, HAMELIN et al., 2014] - "Autour" du smartphone (PLB) [BERGÉ et al., 2015] - DichotoZoom (EB) [BALAA et al., 2018] - Roly-Poly Mouse [PERELMAN et al., 2014; PERELMAN et al., 2015b; PERELMAN et al., 2015a] - Rolling-Menu [DUBOIS et al., 2017; DUBOIS et al., 2018]

personnes non voyantes à effectuer un trajet de manière autonome [KAMMOUN et al., 2012]. Depuis 3 ans, je m'intéresse plus particulièrement aux modalités d'interaction pour faciliter la prise d'informations sur des cartes (ou dessins) en relief [MULET et al., 2020]. Ces travaux ont été facilités grâce au laboratoire commun « Cherchons pour voir » qui a été créé entre l'IRIT et l'institut des jeunes aveugles (IJA) de Toulouse. Ce laboratoire commun permet de concevoir des techniques d'assistance avec l'ensemble des interlocuteurs impliqués dans la problématique de la déficience visuelle [JOUFFRAIS et al., 2014].

Enfin, j'ai également pu participer à d'autres projets sur des thématiques différentes comme la conception d'un outil de lecture adapté pour les personnes dyslexiques [ATHÈNES et al., 2007; ATHÈNES et al., 2009], ou un tableau de bord pour visualiser les données énergétiques du campus [BORTOLASO et al., 2020]. La problématique étant à chaque fois de proposer des techniques d'interaction adaptées au contexte d'interaction.

Perspectives

Comme nous l'avons vu tout au long de ce mémoire, les contextes d'interactions se multiplient et sont de plus en plus variés : là où nous n'avions quasiment que le clavier et la souris pour communiquer avec notre ordinateur, il existe maintenant de nombreux dispositifs d'entrée pour interagir avec les environnements numériques (des écrans tactiles de toutes tailles, des dispositifs bardés de capteurs permettant de capturer de manière très précise les mouvements de l'utilisateur, etc.). De même, les surfaces d'affichages vont du simple cadran de montre à des murs d'écrans, ou à de multiples surfaces d'affichage réparties dans une pièce. Les profils des utilisateurs peuvent être également très variés en fonction de leur expertise du numérique et des technologies, de possibles déficiences (motrice, visuelle, ...), et du contexte dans lequel ils se trouvent (en mobilité, sur un écran très réduit, etc.).

Quel que soit le contexte d'interaction, la saisie de texte, ainsi que le pointage et la manipulation d'éléments, constituent toujours des tâches essentielles à l'interaction avec l'environnement numérique proposé. Cependant, tous ces facteurs font qu'il est de plus en plus difficile d'évaluer un dispositif ou une technique et de le comparer à d'autres solutions proposées dans la littérature.

Deux constats issus de mes travaux viennent en partie expliquer cette difficulté de comparaison entre dispositifs :

1. D'une part, pour pouvoir comparer 2 dispositifs entre eux, il est nécessaire que ces deux dispositifs aient été évalués de manière similaire : c'est-à-dire qu'ils aient été évalués sur des tâches identiques, et que les critères d'évaluation soient les mêmes.
2. D'autre part, vu la variété des contextes d'utilisation, comparer des dispositifs n'a de sens que si ces situations partagent des caractéristiques communes. C'est pourquoi il est donc très important de caractériser un dispositif ainsi que le contexte d'interaction (plus ou moins spécifique) pour lequel il a été conçu ou dans lequel il est utilisé.

Mon premier axe de travail pour les années à venir a pour objectif de formaliser la description des dispositifs et d'autre part de normaliser les pratiques d'évaluation afin d'uniformiser la

conception, l'évaluation et la comparaison des dispositifs. Ce travail est à mener sur les deux thématiques de recherche présentées précédemment, mais à des niveaux différents. Il est dans la continuité de ce que j'ai déjà réalisé avec les étudiants que j'ai encadrés. Certaines parties du travail de formalisation existent déjà, il s'agira donc de combler les manques sur la saisie de texte et le pointage. Comme l'approche est parallèle entre ces deux thématiques, nous pourrions nous appuyer sur nos expériences dans une thématique pour réaliser le même type de travail dans l'autre thématique. Le travail que nous comptons mener sur cet axe de travail est présenté dans la première section.

Au-delà d'unifier la conception et l'évaluation des dispositifs, ce premier travail permettra, grâce aux données normalisées, d'alimenter une base de connaissances nécessaires à notre second axe de travail. En effet, l'axe de recherche que je compte privilégier par la suite consiste à utiliser les algorithmes de machine learning en IHM et plus particulièrement sur mes deux thématiques privilégiées de recherche : la saisie et le pointage. Cet axe de recherche est expliqué dans la seconde section.

Une nécessité de structurer encore et encore ...

Un besoin d'uniformisation des espaces de description

Pour la saisie de texte, nos travaux sur l'espace de description sont une première approche pour essayer d'unifier les différents points de vue et faciliter l'exploration et la comparaison de systèmes de saisie de texte en fonction des critères que nous avons proposés. Nos critères permettent de décrire les différents systèmes d'un point de vue fonctionnel, mais ils définissent également le contexte d'interaction pour lequel ce système a été conçu (dispositif d'interaction utilisée, profil utilisateur etc.).

Pour les techniques de pointage, ce travail de description et de formalisation reste à faire. Il existe de nombreux cadres de conception qui permettent de caractériser différentes solutions de pointage en fonction d'un point de vue donné. Cependant, ces cadres ne décrivent souvent qu'une partie des dispositifs, et qu'en partie les dispositifs : certains vont se focaliser sur un contexte donné (comme nous l'avons fait, par exemple, pour la 3D), d'autres sur les capacités du dispositif, etc.

Un objectif de travail sera donc de proposer un espace de description unifié qui, comme pour la saisie de texte, permettra de décrire les différents dispositifs de pointage à la fois d'un point de vue fonctionnel, mais également en fonction du contexte d'interaction. Ce travail se basera sur une étude approfondie de la littérature, pour en extraire les critères pertinents pour décrire l'ensemble des dispositifs.

Une nécessité de normaliser les évaluations

Concernant la tâche de pointage, la norme ISO 9241-9 fait office de référence pour l'évaluation d'un dispositif de pointage. Cette norme définit la tâche à réaliser, comment la mettre en œuvre, ainsi que les métriques à utiliser pour évaluer les performances d'un dispositif. La norme, publiée en 2012, s'intéressait à la tâche en 2D. Nous avons présenté, à la section [B.1](#) du chapitre 4, notre adaptation de cette norme pour qu'il soit également possible d'évaluer les dispositifs pour une tâche de pointage dans un environnement virtuel en 3D. Cette norme, avec ses adaptations liées au contexte, permet ainsi de proposer une évaluation normalisée pour les dispositifs de pointage.

En revanche, pour la tâche de saisie, bien qu'il existe des métriques des performances des systèmes de saisie, aucune norme n'existe pour évaluer un système de saisie de texte. Bien que les systèmes de saisie de texte soient souvent évalués au travers d'une tâche de recopie de texte, plusieurs paramètres restent souvent à la discrétion de l'évaluateur. En effet, il existe, par exemple, plusieurs manières de prendre en compte les erreurs de saisie de l'utilisateur : certaines évaluations font corriger au fur et à mesure les erreurs de saisie, d'autres laissent les participants saisir librement puis calculent ensuite le taux d'erreurs dans la production écrite. De même, il existe plusieurs métriques pour évaluer la vitesse de saisie de texte d'un utilisateur, son taux d'erreurs lors de la saisie, ou la capacité d'un système de prédiction à proposer la suite de la saisie. La multiplicité des protocoles expérimentaux et des métriques utilisées rend alors compliquée la comparaison de deux dispositifs entre eux.

Un travail est donc nécessaire sur la normalisation des protocoles expérimentaux pour l'évaluation des systèmes de saisie de texte. De la même manière que pour la norme ISO 9241-9 relative à l'évaluation des dispositifs de pointage, le travail consistera à proposer un ensemble de recommandations sur la manière de mettre en œuvre une tâche de saisie qui permette d'évaluer les systèmes de saisie de texte. Ces recommandations concerneront la tâche à réaliser, la manière de prendre en compte les erreurs de saisie, le corpus de texte proposé pour les exercices (à l'image de ce qui se fait dans le domaine des outils du traitement de la langue ou de la parole), et les métriques à calculer pour évaluer les performances du système de saisie de texte évalué.

Pour mettre en œuvre ces recommandations, nous nous appuierons sur une revue systématique de la littérature afin d'identifier l'ensemble des pratiques mises en œuvre pour évaluer des systèmes de saisie de texte. Nous analyserons ensuite les avantages et inconvénients de chaque solution proposée pour déterminer les éléments qui paraissent être les plus pertinents pour une évaluation normalisée de la saisie de texte et qui pourraient faire consensus au sein de la communauté.

Approfondir nos connaissances sur les mécanismes de tâches de saisie

Au-delà des métriques qui nous permettent d'évaluer la capacité d'un dispositif à répondre à la demande généralement formulée ainsi : « réaliser la tâche le plus rapidement possible et le plus précisément possible », il paraît important de comprendre pourquoi l'utilisateur arrive à effectuer une tâche de pointage ou de saisie plus ou moins rapidement, et quels sont les éléments qui peuvent venir interférer pendant sa tâche.

Pour le pointage, il existe de nombreux travaux qui analysent notamment le mouvement de l'utilisateur. En revanche, pour la tâche de saisie, peu de travaux analysent la tâche dans son ensemble, probablement en raison de sa complexité intrinsèque. Les études se focalisent souvent sur une seule sous-tâche (soit de saisie du caractère, soit de sélection de la prédiction de mots ou de caractères) sans forcément regarder l'impact qu'elle peut avoir sur l'exécution de la tâche dans sa globalité. De plus, la tâche de saisie est une tâche beaucoup plus complexe que la tâche de pointage : elle est composée de plusieurs tâches de pointage successives (pour la sélection des caractères ou mots), avec, comme nous l'avons vu précédemment, des évolutions dynamiques au cours de la saisie qui peuvent venir perturber l'utilisateur (par exemple, liste de mots qui évolue au cours de la saisie, ou modification de la disposition des caractères tout au long de la saisie).

Plus que les performances elles-mêmes obtenues à la fin de la tâche de saisie, il sera intéressant d'analyser le comportement de l'utilisateur au cours de la tâche. Pour cela, nous nous intéresserons notamment à l'analyse des mouvements effectués tout au long de la saisie pour étudier la manière dont s'enchaînent les différents pointages. Est-ce « simplement » une succession de pointage ? Ou est-ce que l'utilisateur anticipe les sélections qu'il va faire et par conséquent modifie les sous-tâches de pointage nécessaires ? Au-delà de l'analyse du mouvement, nous nous intéresserons également à l'étude du mouvement oculaire au cours de la tâche, afin d'analyser la manière dont l'utilisateur anticipe la suite de sa saisie sur un clavier logiciel standard, et quels sont les facteurs qui perturbent l'utilisateur quand le clavier est modifié dynamiquement.

L'enjeu est de réussir à diminuer, à terme, les facteurs perturbant l'utilisateur au cours de la saisie, lors de l'intégration des résultats des systèmes de prédiction sur le clavier logiciel. Cela passe donc par une connaissance approfondie des mécanismes moteurs et oculaires que l'utilisateur emploie tout au long de la saisie, pour proposer des aides logicielles qui s'allient au mieux au comportement de l'utilisateur pendant la saisie.

Des outils pour inciter les chercheurs à utiliser ces recommandations

Formaliser la description des dispositifs et normaliser la manière de les évaluer est une première étape nécessaire, mais généralement insuffisante pour que la communauté des cher-

cheurs du domaine prenne en compte ces définitions et recommandations dans la conception et l'évaluation de leurs solutions. Pour améliorer leur prise en compte, une bonne stratégie consiste à mettre à disposition des outils qui facilitent leur utilisation.

Pour la saisie de texte, notre site web recensant les différents systèmes de saisie de texte a pour objectif d'aider à l'exploration et à la conception de nouveaux systèmes de saisie de texte. Ce même type d'outil devra être mis en place dès lors que les critères de description pour les techniques de pointage seront définis.

L'architecture logicielle ainsi que les briques logicielles déjà implémentées ont également pour objectif de faciliter la conception de nouveaux prototypes en réutilisant des composants existants et déjà utilisés. De plus, la plateforme e-assist que j'avais conçue et développée pendant ma thèse, puis que nous avons enrichie avec les travaux de doctorat de Georges Badr et Bruno Merlin a pour vocation d'uniformiser les données recueillies lors d'une évaluation d'un système de saisie de texte. Celle-ci enregistre tous les événements qui se produisent lors d'un exercice de saisie de texte. Ces outils continueront donc à être enrichis avec les résultats issus des travaux présentés précédemment. Nous travaillerons également à une meilleure diffusion de ces outils pour qu'ils puissent être utilisés par le plus grand nombre de collègues.

Intégration des algorithmes de machine learning

Un axe de recherche qu'il me paraît intéressant de développer est l'utilisation des algorithmes de machine learning sur nos données issues des évaluations de systèmes de saisie de texte ou de dispositifs de pointage. A partir d'un ensemble (certes conséquent) de données, les algorithmes de machine learning pourraient apporter plusieurs bénéfices pour l'analyse de nos systèmes et de leur utilisation.

Un apport pour l'évaluation

Le premier avantage serait de pouvoir prédire les performances d'un système de saisie de texte ou d'un dispositif de pointage en fonction des critères caractérisant le nouveau système. Cette prédiction théorique basée sur les données issues de l'ensemble des exemples serait plus précise qu'une évaluation théorique basée sur les modèles prédictifs experts (c'est-à-dire basés sur un modèle mathématique construit par un expert, comme par exemple les lois de Fitts ou Hick-Hyman). En particulier, cela permettrait de prendre en compte des informations qu'il est difficile de considérer habituellement comme la dynamique du mouvement effectué avec les dispositifs. Les réseaux de neurones convolutifs ou récurrents sont connus pour être très efficaces sur ce type de données.

Un atout pour la détection

L'utilisation d'algorithmes d'apprentissage non supervisé permettra, par exemple, de construire automatiquement des profils types d'utilisateurs à partir de nombreuses caractéristiques. De plus, les algorithmes de détections d'anomalie pourraient détecter automatiquement quand une interface n'est pas adaptée à un utilisateur en comparant ses statistiques de performances à celles d'un ensemble d'utilisateurs témoins. Cela permettra, par exemple, d'adapter les aides techniques proposées en fonction des capacités de l'utilisateur. On peut ainsi imaginer adapter la technique de pointage ou le système de saisie de texte en fonction du contexte d'interaction, comme passer d'une situation assise à une situation en mobilité, ou prendre en compte un état de fatigue, ou alors prendre en compte une maladie dégénérative qui réduirait les capacités de l'utilisateur au cours du temps.

Un outil supplémentaire pour la prédiction

De même, l'apprentissage dit supervisé, utilisant des réseaux récurrents, peut permettre de prédire la cible à atteindre ou les caractères à saisir en fonction du mouvement réalisé par l'utilisateur et des actions précédentes. Cette prédiction pourrait donc être une aide complémentaire aux systèmes linguistiques régulièrement utilisés pour prédire la suite de la saisie. Aussi, il sera intéressant d'étudier des techniques de pointage utilisant ces prédictions sur le mouvement pour optimiser les déplacements du curseur et le pointage d'éléments.

Bibliographie

- ADER, L. G. M., B. BOSSAVIT, B. CAULFIELD, M. RAYNAL, K. L. H. TING, J. VANDERDONCKT et N. VIGOUROUX (2019). « HCI challenges in human movement analysis ». In : *IFIP Conference on Human-Computer Interaction*. Springer, Cham, p. 725-730.
- ADER, L. G. M., B. CAULFIELD, B. BOSSAVIT, K. EL RAHEB, M. RAYNAL, N. VIGOUROUX, K. L. H. TING, P. P. IRANI et J. VANDERDONCKT (2020). « Visual user interfaces for human motion ». In : *Proceedings of the International Conference on Advanced Visual Interfaces (AVI'2020)*, p. 1-2.
- ADER, L. G. M. et M. RAYNAL (2020). « Human Movement Analysis for the Design and Evaluation of Interactive Systems and Assistive Devices : Introduction to the Special Thematic Session ». In : *International Conference on Computers Helping People with Special Needs*. Springer, Cham, p. 381-383.
- AL FARAJ, K., M. MOJAHID et N. VIGOUROUX (2009). « Bigkey : A virtual keyboard for mobile devices ». In : *International Conference on Human-Computer Interaction*. Springer, p. 3-10.
- ATHÈNES, S., J.-F. CAMPS, C. JOUFFRAIS, V. ONG-MEANG, M. RAYNAL, P. TRUILLET et J.-L. VINOT (2007). « Evaluation expérimentale de Ysilex, un outil d'aide à la lecture pour les personnes dyslexiques ». In : *SOFMER 2007, 22ème Congrès de la Société Française de Médecine Physique et de Réadaptation*.
- ATHÈNES, S., M. RAYNAL, P. TRUILLET et J.-L. VINOT (2009). « Ysilex-a Friendly reading Interface for Dyslexics. » In : *ICTA 2009, International Conference on Information & Communication Technologies : from Theory to Applications*, p. 15-22.
- AULAGNER, G., R. FRANÇOIS, B. MARTIN, D. MICHEL et M. RAYNAL (2010). « Floodkey : increasing software keyboard keys by reducing needless ones without occultation ». In : *Proceedings of the 10th WSEAS international conference on Applied computer science*. World Scientific, Engineering Academy et Society (WSEAS), p. 412-417.

- BADR, G. (2011). « Modèle théorique et outil de simulation pour une meilleure évaluation des claviers logiciels augmentés d'un système de prédiction de mots ». Thèse de doct. Université de Toulouse, Université Toulouse III-Paul Sabatier.
- BADR, G. et M. RAYNAL (2009a). « Optimized interaction with word prediction list ». In : *Assistive Technology from Adapted Equipment to Inclusive Environments*. IOS Press, p. 871-871.
- BADR, G. et M. RAYNAL (2009b). « WordTree : nouvelle technique d'interaction avec une liste de prédiction ». In : *Proceedings of the 21st International Conference on Association Franco-phone d'Interaction Homme-Machine*, p. 227-233.
- BADR, G. et M. RAYNAL (2009c). « WordTree : results of a word prediction system presented thanks to a tree ». In : *International Conference on Universal Access in Human-Computer Interaction*. Springer, p. 463-471.
- BADR, G. et M. RAYNAL (2010). « Evaluation of wordTree system with motor disabled users ». In : *International Conference on Computers Helping People with Special Needs*. Sous la dir. de K. MIESENBERGER, J. KLAUS, W. ZAGLER et A. KARSHMER. Springer. Berlin, Heidelberg : Springer Berlin Heidelberg, p. 104-111. ISBN : 978-3-642-14100-3.
- BADR, G. et M. RAYNAL (2011). « Centralist ». In : *Everyday Technology for Independence and Care*. IOS Press, p. 944-951.
- BALAA, E., M. RAYNAL, F. CABRIC et E. DUBOIS (2018). « Dichotozoom : Technique de sélection d'objets dans un environnement virtuel 3d affiché sur smartphone ». In : *30eme conférence francophone sur l'interaction homme-machine*, p. 108-118.
- BALAA, E., M. RAYNAL, Y. B. ISSA et E. DUBOIS (2014). « Classification of interaction techniques in the 3d virtual environment on mobile devices ». In : *International Conference on Virtual, Augmented and Mixed Reality*. Springer, p. 3-13.
- BELLMAN, T. et I. S. MACKENZIE (1998). « A probabilistic character layout strategy for mobile text entry ». In : *Graphics Interface*. T. 98, p. 168-176.
- BERGÉ, L.-P. (2015). « Couplage de techniques d'interaction avancées avec des environnements virtuels 3D interactifs ». Thèse de doct. Université Paul Sabatier-Toulouse III.
- BERGÉ, L.-P., E. DUBOIS, M. HOURS-PANCHETTI, M. RAYNAL et C. SANZA (2014). « Towards an engineering approach for advanced interaction techniques in 3D environments ». In : *Workshop HCI Engineering in the sixth ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS 2014)*.
- BERGÉ, L.-P., E. DUBOIS et M. RAYNAL (2015). « Design and Evaluation of an "Around the SmartPhone" Technique for 3D Manipulations on Distant Display ». In : *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, p. 69-78.

- BERGÉ, L.-P., G. PERELMAN, A. HAMELIN, M. RAYNAL, C. SANZA, M. HOURS-PANCHETTI, R. CABANAC et E. DUBOIS (2014). « Smartphone Based 3D Navigation Techniques in an Astronomical Observatory Context : Implementation and Evaluation in a Software Platform ». In : *International Journal on Advances in Software* 7.3-4, p. 551-566.
- BERGÉ, L.-P., G. PERELMAN, M. RAYNAL, C. SANZA, M. SERRANO, M. HOURS-PANCHETTI, R. CABANAC et E. DUBOIS (2014). « Smartphone-based 3D navigation technique for use in a museum exhibit ». In : *The Seventh International Conference on Advances in Computer-Human Interactions (ACHI 2014)*, p. 252-257.
- BERGÉ, L.-P., M. SERRANO, G. PERELMAN et E. DUBOIS (2014). « Exploring smartphone-based interaction with overview+ detail interfaces on 3D public displays ». In : *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. ACM, p. 125-134.
- BLANCH, R., Y. GUIARD et M. BEAUDOUIN-LAFON (2004). « Semantic pointing : improving target acquisition with control-display ratio adaptation ». In : *Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 519-526.
- BORTOLASO, C., S. COMBETTES, M.-P. GLEIZES, B. LARTIGUE, M. RAYNAL et S. REY (2020). « SANDFOX Project Optimizing the Relationship Between the User Interface and Artificial Intelligence to Improve Energy Management in Smart Buildings ». In : *International Conference on Human-Computer Interaction*. Springer, Cham, p. 417-433.
- BOWMAN, D., E. KRUIJFF, J. J. LAVIOLA JR et I. P. POUPYREV (2004). *3D User interfaces : theory and practice, CourseSmart eTextbook*. Addison-Wesley.
- CARD, S. K., W. K. ENGLISH et B. J. BURR (1978). « Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT ». In : *Ergonomics* 21.8, p. 601-613.
- CARD, S. K., T. P. MORAN et A. NEWELL (1980). « The keystroke-level model for user performance time with interactive systems ». In : *Communications of the ACM* 23.7, p. 396-410.
- CASIEZ, G., D. VOGEL, Q. PAN et C. CHAILLOU (2007). « RubberEdge : reducing clutching by combining position and rate control with elastic feedback ». In : *Proceedings of the 20th annual ACM symposium on User interface software and technology*, p. 129-138.
- COCKBURN, A., C. GUTWIN et S. GREENBERG (2007). « A predictive model of menu performance ». In : *Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 627-636.
- DAM, A. van (2001). « Post-Wimp user interfaces : The human connection ». In : *Frontiers of Human-Centered Computing, Online Communities and Virtual Environments*. Springer, p. 163-178.

- DANG, N.-T., H.-H. L. H.-H. LE et M. TAVANTI (2003). « Visualization and interaction on flight trajectory in a 3d stereoscopic environment ». In : *Digital Avionics Systems Conference, 2003. DASC'03. The 22nd*. T. 2. IEEE, 9-A.
- DAS, A. et W. STUERZLINGER (2008). « Modeling learning effects in mobile texting ». In : *Proceedings of the 7th international Conference on Mobile and Ubiquitous Multimedia*, p. 154-161.
- DIAMOND, P. (1988). « Fuzzy least squares ». In : *Information Sciences : an International Journal* 46.3, p. 141-157.
- DRAGICEVIC, P. et J.-D. FEKETE (2001). « Input device selection and interaction configuration with ICON ». In : *People and Computers XV—Interaction without Frontiers*. Springer, p. 543-558.
- DUBE, T. J. et A. S. ARIF (2019). « Text entry in virtual reality : A comprehensive review of the literature ». In : *International Conference on Human-Computer Interaction*. Springer, p. 419-437.
- DUBOIS, E., M. RAYNAL et M. SERRANO (2017). « Taking advantage of rolling gestures on a multidimensionnal mouse for interacting with a menubar ». In : *Proceedings of the 29th Conference on l'Interaction Homme-Machine*, p. 83-92.
- DUBOIS, E., M. SERRANO et M. RAYNAL (2018). « Rolling-menu : rapid command selection in toolbars using roll gestures with a multi-DoF Mouse ». In : *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 1-12.
- FAZLY, A. et G. HIRST (2003). « Testing the efficacy of part-of-speech information in word completion ». In : *Proceedings of the 2003 EACL Workshop on Language Modeling for Text Entry Methods*.
- FITTS, P. M. (1954). « The information capacity of the human motor system in controlling the amplitude of movement. » In : *Journal of experimental psychology* 47.6, p. 381.
- FITTS, P. M. et J. R. PETERSON (1964). « Information capacity of discrete motor responses. » In : *Journal of experimental psychology* 67.2, p. 103.
- FRUCHTERMAN, T. et E. REINGOLD (1991). *Graph Drawing by Force-directed Placement Software-Practice and Experiences*, 21 (11) : 1129-1164.
- GODARD, N., M. RAYNAL, B. MARTIN et J.-L. VINOT (2009). « Etude de l'impact d'une prévisualisation des résultats d'un système de prédiction de caractères ». In : *Proceedings of the 21st International Conference on Association Francophone d'Interaction Homme-Machine*, p. 235-238.

- GROSSMAN, T. et R. BALAKRISHNAN (2006). « The design and evaluation of selection techniques for 3D volumetric displays ». In : *Proceedings of the 19th annual ACM symposium on User interface software and technology*, p. 3-12.
- HICK, W. E. (1952). « On the rate of gain of information ». In : *Quarterly Journal of experimental psychology* 4.1, p. 11-26.
- HINCKLEY, K., R. PAUSCH, J. C. GOBLE et N. F. KASSELL (1994). « A survey of design issues in spatial input ». In : *Proceedings of the 7th annual ACM symposium on User interface software and technology*, p. 213-222.
- HUGHES, D., J. WARREN et O. BUYUKKOKTEN (2002). « Empirical bi-action tables : A tool for the evaluation and optimization of text-input systems. Application I : Stylus keyboards ». In : *Human-Computer Interaction* 17.2-3, p. 271-309.
- HYMAN, R. (1953). « Stimulus information as a determinant of reaction time. » In : *Journal of experimental psychology* 45.3, p. 188.
- ISHII, H., C. WISNESKI, S. BRAVE, A. DAHLEY, M. GORBET, B. ULLMER et P. YARIN (1998). « ambientROOM : integrating ambient media with architectural space ». In : *CHI 98 conference summary on Human factors in computing systems*. ACM, p. 173-174.
- ISO, I. (2000). « 9241-9 Ergonomic requirements for office work with visual display terminals (VDTs)-Part 9 : Requirements for non-keyboard input devices (FDIS-Final Draft International Standard), 2000 ». In : *International Organization for Standardization*.
- JEON, S., J. HWANG, G. J. KIM et M. BILLINGHURST (2010). « Interaction with large ubiquitous displays using camera-equipped mobile phones ». In : *Personal and Ubiquitous Computing* 14.2, p. 83-94.
- JONES, B., R. SODHI, D. FORSYTH, B. BAILEY et G. MACIOCCI (2012). « Around device interaction for multiscale navigation ». In : *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, p. 83-92.
- JOUFFRAIS, C., B. ORIOLA, P. TRUILLET, M. RAYNAL, A. SERPA et M. MACÉ (2014). « Les technologies d'assistance pour la qualité de vie et l'autonomie des déficients visuels (TAVIS) ». In : *8^{eme} Congrès francophone HANDICAP*. IFRATH, p. 307.
- KAMMOUN, S., G. PARSEIHIAN, O. GUTIERREZ, A. BRILHAULT, A. SERPA, M. RAYNAL, B. ORIOLA, M.-M. MACÉ, M. AUVRAY, M. DENIS et al. (2012). « Navigation and space perception assistance for the visually impaired : The NAVIG project ». In : *Innovation and Research in BioMedical engineering* 33.2, p. 182-189.

- KETABDAR, H., M. ROSHANDEL et K. A. YÜKSEL (2010). « MagiWrite : towards touchless digit entry using 3D space around mobile devices ». In : *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, p. 443-446.
- KRATZ, S., M. ROHS, D. GUSE, J. MÜLLER, G. BAILLY et M. NISCHT (2012). « PalmSpace : continuous around-device gestures vs. multitouch for 3D rotation tasks on mobile devices ». In : *Proceedings of the international working conference on advanced visual interfaces*, p. 181-188.
- KRISTENSSON, P.-O. et S. ZHAI (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*, p. 43-52.
- LAVIOLA, J. J. et D. F. KEEFE (2011). « 3D spatial interaction : applications for art, design, and science ». In : *ACM Siggraph 2011 Courses*. ACM, p. 1.
- LEE, D., J.-I. HWANG, G. J. KIM et S. C. AHN (2011). « 3D interaction using mobile device on 3D environments with large screen ». In : *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. ACM, p. 575-580.
- LIANG, J. et M. GREEN (1993). « Geometric modeling using six degrees of freedom input devices ». In : *Proc. 3rd International Conference on CAD and Computer Graphics*. Citeseer, p. 217-222.
- LUNA, M. M., F. A. A. de MELO NUNES, H. A. D. do NASCIMENTO, J. SIQUEIRA, E. F. de SOUZA, T. H. NASCIMENTO, R. M. da COSTA et al. (2018). « Text entry on smartwatches : A systematic review of literature ». In : *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. T. 2. IEEE, p. 272-277.
- MACKENZIE, I. S. (2002). « KSPC (keystrokes per character) as a characteristic of text entry techniques ». In : *International Conference on Mobile Human-Computer Interaction*. Springer, p. 195-210.
- MACKENZIE, I. S. et B. ASHTIANI (2011). « BlinkWrite : efficient text entry using eye blinks ». In : *Universal Access in the Information Society* 10.1, p. 69-80.
- MACKENZIE, I. S. et W. BUXTON (1992). « Extending Fitts' law to two-dimensional tasks ». In : *Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 219-226.
- MACKENZIE, I. S. et R. W. SOUKOREFF (2002). « Text entry for mobile computing : Models and methods, theory and practice ». In : *Human-Computer Interaction* 17.2-3, p. 147-198.
- MACKENZIE, I. S. et R. W. SOUKOREFF (2003). « Phrase sets for evaluating text entry techniques ». In : *CHI'03 extended abstracts on Human factors in computing systems*, p. 754-755.

- MAGLIOCCHETTI, D., G. CONTI et R. DE AMICIS (2012). « I-MOVE : towards the use of a mobile 3D GeoBrowser framework for urban mobility decision making ». In : *International Journal on Interactive Design and Manufacturing (IJIDeM)* 6.4, p. 205-214.
- MAGNIEN, L., J. L. BOURAOUI et N. VIGOUROUX (2004). « Mobile text input with soft keyboards : optimization by means of visual clues ». In : *International Conference on Mobile Human-Computer Interaction*. Springer, p. 337-341.
- MARTIN, B. et I. PECCI (2007). « État de l'art des claviers physiques et logiciels pour la saisie de texte ». In : *Revue d'Interaction Homme-Machine (RIHM)* 2.
- MARTINET, A., G. CASIEZ et L. GRISONI (2010). « The design and evaluation of 3d positioning techniques for multi-touch displays ». In : *2010 IEEE symposium on 3D user interfaces (3DUI)*. IEEE, p. 115-118.
- MARTON, F., M. B. RODRIGUEZ, F. BETTIO, M. AGUS, A. J. VILLANUEVA et E. GOBBETTI (2014). « IsoCam : Interactive visual exploration of massive cultural heritage models on large projection setups ». In : *Journal on Computing and Cultural Heritage (JOCCH)* 7.2, p. 1-24.
- MASUI, T. (1999). « POBox : An efficient text input method for handheld and ubiquitous computers ». In : *International Symposium on Handheld and Ubiquitous Computing*. Springer, p. 289-300.
- MAUREL, F., M. RAYNAL, B. ORIOLA et N. VIGOUROUX (s. d.). « Extension of W3C Standards for Real Web Accessibility ». In : *2nd International Conference on Universal Access in Human-Computer Interaction*, p. 995-999.
- MAUREL, F., N. VIGOUROUX, M. RAYNAL et B. ORIOLA (2003). « Contribution of the Transmodality Concept to Improve Web Accessibility ». In : *Assistive Technology Research Series* 12, p. 186-193.
- MERLIN, B. et M. RAYNAL (2009). « SpreadKey : increasing software keyboard key by recycling needless ones ». In : *10th European conference for the advancement of assistive technology in europe (AAATE 2009)*, p. 138-143.
- MERLIN, B. (2011). « Méthodologie et instrumentalisation pour la conception et l'évaluation des claviers logiciels ». Thèse de doct. Université de Toulouse, Université Toulouse III-Paul Sabatier.
- MERLIN, B., C. HURTER et M. RAYNAL (2009). « Bridging software evolution's gap : the multi-layer concept ». In : *International Conference on Human Centered Design*. Springer, Berlin, Heidelberg, p. 266-275.
- MERLIN, B. et M. RAYNAL (2010). « Evaluation of SpreadKey system with motor impaired users ». In : *International Conference on Computers for Handicapped Persons*. Springer, p. 112-119.

- MERLIN, B. et M. RAYNAL (2012). « Soft Keyboard evaluations : Integrating user's background in predictive models ». In : *Cognitively Informed Intelligent Interfaces : Systems Design and Development*. IGI Global, p. 21-40.
- MERLIN, B., M. RAYNAL et H. FÜLBER (2012a). « E-Assist II : a platform to design and evaluate soft-keyboards ». In : *Workshop on Designing and Evaluating Text Entry Methods, ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'12)*.
- MERLIN, B., M. RAYNAL et H. FÜLBER (2012b). « Multilayer Keyboard : transition toward a new optimized layout ». In : *Workshop on Designing and Evaluating Text Entry Methods, ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'12)*.
- MERLIN, B., M. RAYNAL et H. FÜLBER (2012c). « Tackling the acceptability of freely optimized keyboard layout ». In : *International Conference on Computers for Handicapped Persons*. Springer, Berlin, Heidelberg, p. 459-466.
- MULET, J., L. DIMITROV, A. BARTOLUCCI, M. RAYNAL, V. TARTAS, J. DUCASSE, M. J.-M. MACÉ, B. ORIOLA, J. LEMARIÉ et C. JOUFFRAIS (2020). « Évaluation d'un dispositif audio-tangible pour l'apprentissage spatial chez les enfants déficients visuels ». In : *11ème conférence de l'IFRATH sur les technologies d'assistance : Technologies pour l'autonomie et l'inclusion (Handicap 2020)*. IFRATH, p. 157-162.
- NANCEL, M., O. CHAPUIS, E. PIETRIGA, X.-D. YANG, P. P. IRANI et M. BEAUDOUIN-LAFON (2013). « High-precision pointing on large wall displays using small handheld devices ». In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 831-840.
- NIGAY, C. (1997). *The CARE Properties and Their Impact on Software Design. Intelligence and Multimodality in Multimedia Interfaces : Research and Applications*, John Lee.
- ORIOLA, B., F. AMADIEU, P. BOISSIÈRE, M. RAYNAL, A. TRICOT, F. VELLA et N. VIGOUROUX (2007). « Projet ChatCom : Étude des usages de différents claviers logiciels dans une tâche de saisie via Internet par des personnes handicapées ». In : *1ère Conférence Internationale sur l'accessibilité et les systèmes de suppléance aux personnes en situations de handicaps (ASSISTH 2007)*. IRIT : Institut de Recherche en Informatique de Toulouse. Cépaduès, Toulouse, p. 233-240.
- PAVLOVYCH, A. et W. STUERZLINGER (2004). « Model for non-expert text entry speed on 12-button phone keypads ». In : *Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 351-358.
- PECCHIOLO, L., M. CARROZZINO, F. MOHAMED, M. BERGAMASCO et T. H. KOLBE (2011). « ISEE : Information access through the navigation of a 3D interactive environment ». In : *Journal of Cultural Heritage* 12.3, p. 287-294.

- PERELMAN, G., M. SERRANO, M. RAYNAL, C. PICARD, M. DERRAS et E. DUBOIS (2015a). « The roly-poly mouse : Designing a rolling input device unifying 2d and 3d interaction ». In : *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, p. 327-336.
- PERELMAN, G., M. SERRANO, M. RAYNAL, C. PICARD, M. DERRAS et E. DUBOIS (2016). « DECO : A design space for device composition ». In : *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, p. 435-446.
- PERELMAN, G., M. SERRANO, M. RAYNAL, C. PICARD, M. DERRAS et E. DUBOIS (2014). « Conception d'un Dispositif pour Interagir avec des Données Multidimensionnelles : Disco ». In : *IHM'14, 26e conférence francophone sur l'Interaction Homme-Machine*. ACM, p. 91-100.
- PERELMAN, G., M. SERRANO, M. RAYNAL, C. PICARD, M. DERRAS et E. DUBOIS (2015b). « Disco (s) : Dispositifs à Multiples Degrés de Liberté pour Interagir avec des Données Multidimensionnelles ». In : *27ème conférence francophone sur l'Interaction Homme-Machine*. P. d06.
- POIRIER, F., C. KOLSKI et Y. GUERRIER (2018). « State of art of possible communication support systems for users with athetoid cerebral palsy ». In : *Journal d'Interaction Personne-Système* 6.
- POLLMANN, F., D. WENIG et R. MALAKA (2014). « HoverZoom : making on-screen keyboards more accessible ». In : *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, p. 1261-1266.
- RAYNAL, M. (2004). « KeyGlasses : Des touches semi-transparentes pour optimiser la saisie de texte ». In : *Rencontres Jeunes Chercheurs en Interaction Homme-Machine*.
- RAYNAL, M. (2005). « Systemes de saisie de textes pour les personnes handicapées moteur : optimisation, interaction et mesure de l'utilisabilité ». Thèse de doct. Université de Toulouse, Université Toulouse III-Paul Sabatier.
- RAYNAL, M. (2006). « Claviers GAG : claviers logiciels optimisés pour la saisie de texte au stylet ». In : *Proceedings of the 18th Conference on l'Interaction Homme-Machine*, p. 3-10.
- RAYNAL, M. (2007). « Le système KEYGLASS : Système d'ajout dynamique de touches sur clavier logiciel ». In : *Traitement Automatique des Langues* 48.2, p. 97-121.
- RAYNAL, M. (2014). « KeyGlasses : semi-transparent keys on soft keyboard ». In : *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*, p. 347-349.
- RAYNAL, M., E. DUBOIS et B. SCHMITT (2013). « Towards unification for pointing task evaluation in 3D desktop virtual environment ». In : *International Conference on Human Factors in Computing and Informatics*. Springer, p. 562-580.

- RAYNAL, M., G. GAUFFRE, C. BACH, B. SCHMITT et E. DUBOIS (2010). « Tactile camera vs. tangible camera : Taking advantage of small physical artefacts to navigate into large data collection ». In : *Proceedings of the 6th Nordic Conference on Human-Computer Interaction : Extending Boundaries*, p. 373-382.
- RAYNAL, M., I. S. MACKENZIE et B. MERLIN (2014). « Semantic keyboard : Fast movements between keys of a soft keyboard ». In : *International Conference on Computers for Handicapped Persons*. Springer, p. 195-202.
- RAYNAL, M. et B. MARTIN (2020). « SlideKey : Impact of In-depth Previews for a Predictive Text Entry Method ». In : *International Conference on Computers Helping People with Special Needs*. Springer, p. 363-370.
- RAYNAL, M., S. MAUBERT, N. VIGOUROUX, F. VELLA et L. MAGNIEN (2005). « E-Assiste : A Platform Allowing Evaluation of Text Input Systems ». In : *3rd Int. Conf. on Universal Access in Human-Computer Interaction (UAHCI 2005)*. Las vegas, United States.
- RAYNAL, M. et P. ROUSSILLE (2017). « DUCK : A DeDUctive Soft Keyboard for Visually Impaired Users. » In : *Studies in health technology and informatics* 242, p. 902-909.
- RAYNAL, M. et P. ROUSSILLE (2018). « Use of DUCK Keyboard for a Daily Use ». In : *International Conference on Computers Helping People with Special Needs*. Springer, p. 399-406.
- RAYNAL, M. et M. SERRURIER (2002). « CYNTHIA : An HTML browser for visually handicapped people ». In : *International Conference on Computers for Handicapped Persons*. Springer, Berlin, Heidelberg, p. 353-359.
- RAYNAL, M. et M. SERRURIER (2007). « Loi de fitts : prédiction du temps de pointage sous forme d'ensembles flous ». In : *Proceedings of the 19th Conference on l'Interaction Homme-Machine*, p. 3-10.
- RAYNAL, M. et P. TRUILLET (2006). « Le clavier fisheye ». In : *Proceedings of the 18th Conference on l'Interaction Homme-Machine*, p. 289-290.
- RAYNAL, M. et P. TRUILLET (2007). « Fisheye keyboard : Whole keyboard displayed on PDA ». In : *International Conference on Human-Computer Interaction*. Springer, Berlin, Heidelberg, p. 452-459.
- RAYNAL, M. et N. VIGOUROUX (2005a). « Genetic algorithm to generate optimized soft keyboard ». In : *CHI'05 extended abstracts on Human factors in computing systems*, p. 1729-1732.
- RAYNAL, M. et N. VIGOUROUX (2005b). « KeyGlasses : Semi-transparent keys to optimize text input on virtual keyboard ». In : *8th European conference for the advancement of assistive technology in europe (AAATE 2005)*.

- RAYNAL, M., J.-L. VINOT et P. TRUILLET (2007). « Fisheye keyboard : whole keyboard displayed on small device ». In : *UIST 2007, 20th annual ACM Symposium on User Interface Software and Technology*, pp-65.
- ROUSSILLE, P. et M. RAYNAL (2016a). « LOVIE : a word list optimized for visually impaired Users on smartphones ». In : *International Conference on Universal Access in Human-Computer Interaction*. Springer, p. 185-197.
- ROUSSILLE, P. et M. RAYNAL (2016b). « SWIFT : a short word solution for fast typing ». In : *International Conference on Computers Helping People with Special Needs*. Springer, p. 464-471.
- ROUSSILLE, P., M. RAYNAL et C. JOUFFRAIS (2015). « Clavier DUCK : Utilisation d'un système de déduction de mots pour faciliter la saisie de texte sur écran tactile pour les non-voyants ». In : *27ème conférence francophone sur l'Interaction Homme-Machine*. ACM, a19.
- ROUSSILLE, P., M. RAYNAL, S. KAMMOUN, E. DUBOIS et C. JOUFFRAIS (2013). « DUCK : a de-DUCTive Keyboard ». In : *3rd International Workshop on Mobile Accessibility (MOBACC 2013)*. ACM, pp-1.
- SAD, H. H. et F. POIRIER (2009). « Modeling word selection in predictive text entry ». In : *International Conference on Human-Computer Interaction*. Springer, p. 725-734.
- SCHMITT, B., E. DUBOIS, M. RAYNAL, C. BACH et D. CROENNE (2011). « Evaluation d'une technique d'interaction ubiquitaire pour le pointage de données complexes et spatialisées ». In : *Mobilité et Ubiquité - UBIMOB'11*.
- SCHMITT, B., M. RAYNAL, C. BACH, D. CROENNE et E. DUBOIS (2013). « Vers une unification de l'évaluation de la tâche de pointage en environnement virtuel 3D ». In : *Technique et Science Informatiques* 32.5, p. 605-638.
- SCHMITT, B., M. RAYNAL et E. DUBOIS (2012). « Poster : A composite approach to evaluate two interaction techniques for a 3D pointing task ». In : *2012 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, p. 159-160.
- SCHMITT, B., M. RAYNAL, E. DUBOIS et C. BACH (2011). « Extension de la norme ISO 9241-9 au pointage en 3D ». In : *Proceedings of the 23rd Conference on l'Interaction Homme-Machine*, p. 1-4.
- SEARS, A., J. A. JACKO, J. CHU et F. MORO (2001). « The role of visual search in the design of effective soft keyboards ». In : *Behaviour & Information Technology* 20.3, p. 159-166.
- SERRURIER, M. et H. PRADE (2006). « Imprecise regression and regression on fuzzy data A preliminary discussion ». In : *the 2006 IEEE International Conference on Fuzzy Systems*. IEEE, p. 73-83.

- SERRURIER, M. et M. RAYNAL (2008). « Learning Fitts' law with imprecise regression ». In : *NAFIPS 2008-2008 Annual Meeting of the North American Fuzzy Information Processing Society*. IEEE, p. 1-6.
- SILFVERBERG, M., I. S. MACKENZIE et P. KORHONEN (2000). « Predicting text entry speed on mobile phones ». In : *Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 9-16.
- SIQUEIRA, J., F. A. A. de MELO SOARES, D. J. FERREIRA, C. R. G. SILVA, L. de OLIVEIRA BERRETTA, C. B. R. FERREIRA, I. M. FÉLIX, A. da SILVA SOARES, R. M. da COSTA et M. M. LUNA (2016). « Braille text entry on smartphones : A systematic review of the literature ». In : *2016 IEEE 40th annual computer software and applications conference (COMPSAC)*. T. 2. IEEE, p. 521-526.
- SONG, P., W. B. GOH, C.-W. FU, Q. MENG et P.-A. HENG (2011). « WYSIWYF : exploring and annotating volume data with a tangible handheld device ». In : *Proceedings of the SIGCHI conference on human factors in computing systems*, p. 1333-1342.
- SOUKOREFF, R. W. et I. S. MACKENZIE (2004). « Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI ». In : *International journal of human-computer studies* 61.6, p. 751-789.
- SOUKOREFF, W. R. et I. S. MACKENZIE (1995). « Theoretical upper and lower bounds on typing speed using a stylus and a soft keyboard ». In : *Behaviour & Information Technology* 14.6, p. 370-379.
- TANAKA, H. (1987). « Fuzzy data analysis by possibilistic linear models ». In : *Fuzzy sets and systems* 24, p. 363-376.
- TEATHER, R. J. et W. STUERZLINGER (2013). « Pointing at 3d target projections with one-eyed and stereo cursors ». In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 159-168.
- TELKENAROGLU, C. et T. CAPIN (2013). « Dual-finger 3d interaction techniques for mobile devices ». In : *Personal and ubiquitous computing* 17.7, p. 1551-1572.
- TRUILLET, P. et M. RAYNAL (2013). « Conception rapide d'aides techniques pour personnes tétraplégiques ». In : *Médiation et Information* 36, p. 103-115.
- TRUILLET, P., N. VIGOUROUX, A. SERPA, M. RAYNAL et F. VELLA (2019). « Démarche de co-conception centrée utilisateur des systèmes d'assistance : la place de l'écosystème ». In : *Journée IHM & Santé*.

- TSANDILAS, T., E. DUBOIS et M. RAYNAL (2010). « Free-space pointing with constrained hand movements ». In : *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, p. 3451-3456.
- TSANDILAS, T., E. DUBOIS et M. RAYNAL (2013). « Modeless pointing with low-precision wrist movements ». In : *IFIP Conference on Human-Computer Interaction*. Springer, p. 494-511.
- VAJK, T., P. COULTON, W. BAMFORD et R. EDWARDS (2008). « Using a mobile phone as a “wii-like” controller for playing games on a large public display ». In : *International Journal of Computer Games Technology* 2008.
- VIGOUROUX, N., F. VELLA, M. RAYNAL et P. BOISSIÈRE (2005). « Solutions et défis pour une meilleure accessibilité et utilisabilité des communicateurs : Optimisation de la saisie de texte ». In : *Actes des entretiens de la fondation de garches : Handicap et environnement : de l'adaptation du logement à l'accessibilité de la cité*, p. 209-22.
- VIGOUROUX, N., F. VELLA, P. TRUILLET et M. RAYNAL (2004). « Evaluation of AAC for text input by two groups of subjects : able-bodied subjects and disabled motor subjects ». In : *8th ERCIM Workshop, User Interface for All, Vienne, Autriche*, p. 28-29.
- WANDMACHER, T., J.-Y. ANTOINE, F. POIRIER et J.-P. DÉPARTE (2008). « Sibylle, an assistive communication system adapting to the context and its user ». In : *ACM Transactions on Accessible Computing (TACCESS)* 1.1, p. 1-30.
- WARD, D. J., A. F. BLACKWELL et D. J. MACKAY (2000). « Dasher-a data entry interface using continuous gestures and language models ». In : *UIST*. Citeseer, p. 129-137.
- WILLIAMS, C., X. D. YANG, G. PARTRIDGE, J. MILLAR-USISKIN, A. MAJOR et P. IRANI (2011). « TZee : exploiting the lighting properties of multi-touch tabletops for tangible 3d interactions ». In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, p. 1363-1372.
- WOBBROCK, J. O. (2006). *EdgeWrite : A versatile design for text entry and control*.
- WOBBROCK, J. O., B. A. MYERS et J. A. KEMBEL (2003). « EdgeWrite : a stylus-based text entry method designed for high accuracy and stability of motion ». In : *Proceedings of the 16th annual ACM symposium on User interface software and technology*, p. 61-70.
- WOODWORTH, R. S. (1899). « Accuracy of voluntary movement. » In : *The Psychological Review : Monograph Supplements* 3.3, p. i.
- ZADEH, L. A. (1965). « Fuzzy sets, Information and Control, vol. 8 ». In : *Google Scholar Google Scholar Digital Library Digital Library*, p. 338-353.

ZHAI, S. (1995). « Human Performance in Six Degree of Freedom Input Control ». Thèse de doct. University of Toronto.

ZHAI, S., W. BUXTON et P. MILGRAM (1994). « The “Silk Cursor” investigating transparency for 3D target acquisition ». In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 459-464.

ZHAI, S., M. HUNTER et B. A. SMITH (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*, p. 119-128.

ZHAI, S., A. SUE et J. ACCOT (2002). « Movement model, hits distribution and learning in virtual keyboarding ». In : *Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 17-24.