

THESE
Présentée par
Gilles Bailly

Pour obtenir le titre de
DOCTEUR de l'UNIVERSITE JOSEPH FOURIER – GRENOBLE 1
(arrêtés ministériel du 5 juillet 1984 et du 30 mars 1992)
Spécialité : Informatique

Techniques de menus : **Caractérisation, Conception et Evaluation**

Soutenue le 26 mai 2009 devant le jury composé de

Président :	Franck Poirier
Directeur de thèse :	Laurence Nigay
Co-directeur de thèse :	Eric Lecolinet
Rapporteurs :	Monique Noirhomme Michel Beaudouin-Lafon
Examineurs :	Christophe Chaillou Pierre Dragicevic

Thèse préparée au sein du Laboratoire d'Informatique de Grenoble (LIG)
385, rue de la Bibliothèque, BP 53, 38041 GRENOBLE CEDEX 9
Université Joseph Fourier – Grenoble 1

Et à Télécom ParisTech, département Informatique et Réseaux,
46 rue Barrault, F-75634 Paris Cedex 13

Remerciements

Je tiens tout d'abord à remercier Laurence Nigay et Eric Lecolinet avec qui ce fut un plaisir de travailler durant cette thèse et qui ont su me motiver quand il le fallait.

Merci à Nicolas Thierry-Mieg qui m'a donné, en premier, goût à la recherche.

Merci à David Auber pour son aide durant mon Master Recherche.

Je remercie les membres de mon jury pour avoir lu attentivement mes travaux, pour s'être déplacés parfois de loin, de m'avoir écouté et pour m'avoir fait part de remarques constructives dans leurs rapports.

Je tiens ensuite à remercier toute l'équipe IHM, Joëlle chef de l'équipe, Céline, JS, Renaud, Marcos, Michael, Gaëlle et tous les autres. Je tiens également à remercier l'équipe de Télécom ParisTech, Anne, Hayette, Tahir, Billel, Sylvain M., Mathias, Aurélien et Sylvain F..

Un grand merci à Marie-Pierre de m'avoir soutenu tout au long de cette thèse et en particulier lors de la rédaction ^^, elle m'a permis de garder le moral.

Merci à tous mes amis.

Merci à tous ceux qui ont accepté de participer aux différentes expériences utilisateurs.

Enfin, merci à mes parents.

Table des matières

Chapitre 1 Introduction

<u>1</u>	CONTEXTE	3
<u>2</u>	CONTRIBUTIONS	4
2.1	CARACTERISATION DES MENUS	4
2.2	CONCEPTION ET EVALUATION DE MENUS	4
<u>3</u>	ORGANISATION DU MEMOIRE	5

Partie I CARACTERISATION DES TECHNIQUES DE MENUS
--

Chapitre 2 Définitions, concepts et terminologies

<u>1</u>	MENUS	13
1.1	DEFINITIONS	13
1.2	CLASSIFICATIONS	19
<u>2</u>	MENU D'UN POINT DE VUE SYSTEME	21
2.1	MODE NOVICE	21
2.2	MODE EXPERT	22
2.3	MODE INTERMEDIAIRE	22
<u>3</u>	MENU D'UN POINT DE VUE UTILISATEUR	23
3.1	EXPERTISE DE L'UTILISATEUR	23
3.2	PRECISION DU BUT DE L'UTILISATEUR	24
<u>4</u>	MENU D'UN POINT DE VUE INTERACTION	26
4.1	CHEMIN FORTEMENT DETOURNE	27
4.2	CHEMIN FAIBLEMENT DETOURNE	27
4.3	CHEMIN DROIT	28
4.4	CHEMIN DIRECT	29
<u>5</u>	SYNTHESE	30

Chapitre 3 Utilisabilité et Applicabilité des menus : MenUA

1	ESPACE DE CONCEPTION	34
1.1	CRITERES DE CONCEPTION	34
1.2	MODELES DE PERFORMANCES	35
1.3	TAXONOMIE DES MENUS 3D	39
1.4	SYNTHESE	41
2	MENUA : STRUCTURE GLOBALE	42
2.1	DESCRIPTION	43
2.2	FACTEUR D'UTILISABILITE	43
2.3	FACTEUR D'APPLICABILITE	45
3	CONCLUSION	47

Chapitre 4 Utilisabilité des techniques de menus

1	RAPIDITE ET PRECISION	52
1.1	ACTIVATION DU MENU	53
1.2	RECHERCHE VISUELLE ET PRISE DE DECISION	53
1.3	SELECTION D'UN ITEM	58
1.4	ACTIVATION D'UN ITEM	66
1.5	MODE EXPERT	66
2	APPRENTISSAGE ET MEMORISATION	68
2.1	MEMOIRE	68
2.2	APPRENTISSAGE DE LA TECHNIQUE	70
2.3	APPRENTISSAGE ET MEMORISATION DE L'ORGANISATION	71
2.4	APPRENTISSAGE ET MEMORISATION DU MODE EXPERT	73
3	SATISFACTION	76
3.1	EXPERIENCE UTILISATEUR	77
3.2	EMOTION	77
3.3	SATISFACTION ET MENUS	78
4	CONCLUSION	80

Chapitre 5 Applicabilité des techniques de menus

<u>1</u>	<u>ADEQUATION A L'APPLICATION</u>	84
1.1	NOMBRE D'ITEMS EN LARGEUR	84
1.2	PROFONDEUR DU SYSTEME DE MENUS	89
1.3	NOMBRE TOTAL D'ITEMS	90
<u>2</u>	<u>ADEQUATION A LA PLATEFORME</u>	92
2.1	DISPOSITIFS D'ENTREE	92
2.2	CONTRAINTES SPATIALES	95
<u>3</u>	<u>ADEQUATION A LA TACHE</u>	97
3.1	CONTEXTE VISUEL	97
3.2	TRANSITIONS	99
<u>4</u>	<u>CONCLUSION</u>	104

Synthèse de la partie I 105

Partie II CONCEPTION ET EVALUATION DES TECHNIQUES DE MENUS
--

Chapitre 6 Exploration dans les systèmes de menus : les Wave menus, une solution pour les Marking menus hiérarchiques

<u>1</u>	<u>EXPLORATION DANS LES MENUS</u>	114
1.1	EXPLORATION DANS LES SYSTEMES D'INFORMATION	114
1.2	TACHE DE FEUILLETAGE	116
1.3	TACHE DE RECHERCHE	116
1.4	RECHERCHE VISUELLE	119
1.5	SOLUTIONS CONCEPTUELLES POUR L'EXPLORATION	122
<u>2</u>	<u>WAVE MENUS</u>	126
2.1	RAPPEL DES LIMITATIONS DES MARKING MENUS HIERARCHIQUES	126
2.2	PRINCIPE DES WAVE MENUS	127
2.3	PROPRIETES DES WAVE MENUS	129
2.4	WAVE MENUS INVERSEES	136

3	<u>EVALUATION</u>	137
3.1	DIFFICULTES DE L'EVALUATION POUR LES CHEMINS DETOURNES	137
3.2	DEFINITION ET REALISATION D'UNE TACHE DE NAVIGATION	138
3.3	EXPERIENCE	140
3.4	RESULTATS	142
3.5	DISCUSSIONS	144
4	<u>CONCLUSION</u>	146

Chapitre 7 De la discrimination à l'apprentissage des gestes : les Flower menus, une solution pour les Marking menus

1	<u>ORGANISATION DES MENUS</u>	150
1.1	BALANCEMENT LARGEUR VS. PROFONDEUR	150
1.2	FORME DES MENUS	153
1.3	APPLICATIONS EXISTANTES	154
1.4	CONCLUSION : LE CAS DES MARKING MENUS	156
2	<u>GESTES DES MARKING MENUS</u>	156
2.1	NOMBRE MAGIQUE 7	156
2.2	GESTES DES MARKING MENUS	157
2.3	INTRODUCTION DE LA POSITION RELATIVE (ZONE ET POLYGON MENUS)	157
2.4	INTRODUCTION DE LA COURBURE ET DU SENS	158
3	<u>EVALUATION DES GESTES COURBES</u>	161
3.1	EXPERIMENTATION	161
3.2	RESULTATS	163
3.3	SYSTEME DE RECONNAISSANCE GESTUEL	164
4	<u>FLOWER MENUS</u>	165
4.1	LIMITATIONS DES MARKING MENUS	166
4.2	PRINCIPE DES FLOWER MENUS	167
4.3	PROPRIETES DES FLOWER MENUS	169
5	<u>EVALUATION</u>	176
5.1	EXPERIENCES EXISTANTES	176
5.2	EXPERIMENTATION	178
5.3	RESULTATS	181
5.4	DISCUSSIONS	183
5.5	APPRENTISSAGE ET MEMORISATION : PISTES DE REFLEXION	185
6	<u>CONCLUSION</u>	188

Chapitre 8 Des petites aux grandes surfaces interactives : les Leaf menus et les Multi-Touch Menus (MTM)

1	<u>LEAF MENU : UN MENU POUR DISPOSITIFS MOBILES</u>	192
1.1	INTERACTION SUR DISPOSITIFS MOBILES	192
1.2	MENUS POUR DISPOSITIFS MOBILES	193
1.3	PRINCIPE DU LEAF MENU	194
1.4	PROPRIETES INTERACTIONNELLES DU LEAF MENU	196
1.5	EVALUATION PRELIMINAIRE	199
1.6	SYNTHESE	203
2	<u>MULTI-TOUCH MENU : UN MENU POUR SURFACES MULTI-POINTS</u>	203
2.1	DEGRES DE LIBERTE DE LA MAIN	203
2.2	SURFACES INTERACTIVES MULTI-POINTS	206
2.3	TECHNIQUES DE MENUS MULTI-POINTS	209
2.4	MULTI-TOUCH MENU (MTM)	212
2.5	PROPRIETES INTERACTIONNELLES DU MULTI-TOUCH MENU	216
2.6	SYNTHESE	222
3	<u>CONCLUSION</u>	223

Chapitre 9 Conclusion

1	<u>CONTRIBUTIONS DE LA THESE</u>	227
1.1	CARACTERISATION DES MENUS : ESPACE MENUA	227
1.2	NOUVELLES TECHNIQUES DE MENUS	227
2	<u>PERSPECTIVES A MOYEN TERME</u>	229
2.1	MENUA	229
2.2	PLATEFORME POUR LA REALISATION DE MENUS	230
3	<u>PERSPECTIVES A LONG TERME</u>	233

Table des Figures 235

Bibliographie 247

Annexe

Introduction

Chapitre 1

1	CONTEXTE	3
2	CONTRIBUTIONS	4
2.1	CARACTERISATION DES MENUS	4
2.2	CONCEPTION ET EVALUATION DE MENUS	4
2.2.1	CONCEPTION DE QUATRE NOUVELLES TECHNIQUES DE MENUS	4
2.2.2	EVALUATION DE MENUS	5
3	ORGANISATION DU MEMOIRE	5

1 Contexte

La sélection de commandes est au cœur de l'Interaction Homme-Machine (IHM) car elle permet à l'utilisateur d'accéder aux fonctionnalités proposées par le système interactif. Les menus sont les principaux composants graphiques pour organiser, présenter et sélectionner les commandes. Ils sont apparus en 1968 avec le système AMBIT/G (élaboré dans les laboratoires du MIT) et ont été répandus par le Xerox Star en 1981 et le Macintosh d'Apple en 1984 [Myers 98]. Les menus sont aujourd'hui incontournables dans les interfaces graphiques et doivent permettre de contenir un nombre sans cesse grandissant de commandes.

Les menus ont concouru à la démocratisation de l'informatique car leur manipulation est simple comparée aux langages de commandes. Ils sont utilisés par des utilisateurs variés : de l'utilisateur occasionnel qui découvre les services proposés par une nouvelle application à l'utilisateur expérimenté (par exemple, un graphiste 3D) qui exécute des centaines de commandes par jour. Ainsi, les menus concernent un large public et, une amélioration, même limitée, peut augmenter la productivité et le confort d'utilisation de manière significative.

Cet enjeu a motivé de nombreux travaux en IHM sur les menus. Depuis 25 ans, nous observons un foisonnement de nouvelles techniques de menus conçues pour améliorer les traditionnels menus linéaires avec leurs raccourcis claviers. Par exemple, les *Marking menus* [Kurtenbach 91], illustrés à la Figure 1, constituent une alternative qui facilite l'apprentissage du mode expert en combinant les avantages des menus avec ceux de l'interaction gestuelle [Kurtenbach 93b, Buxton 95, Moyle 02, Bau 08]. En mode novice, les utilisateurs peuvent explorer les commandes dans un menu circulaire qui apparaît après un délai. En mode expert, ils n'attendent pas ce délai et peuvent dessiner la marque associée à la commande sans que le menu apparaisse. Le principal avantage de cette technique est d'accompagner l'apprentissage du mode expert sans effort car l'utilisateur fait le même geste lorsqu'il sélectionne la commande dans le menu où lorsqu'il dessine la marque. C'est une différence essentielle avec les raccourcis claviers qui sont difficiles à apprendre car ils forcent l'utilisateur à changer radicalement de style d'interaction et sont de ce fait assez peu utilisés, même par les utilisateurs expérimentés [Lan 05].

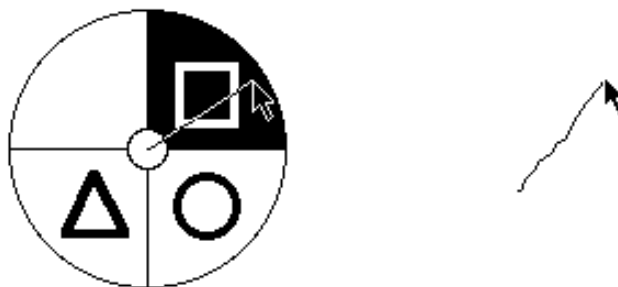


Figure 1 : Marking menus classiques [Kurtenbach 91]. A gauche, le menu circulaire apparaît après un délai de 300ms. Si l'utilisateur n'attend pas ce délai, il sélectionne la commande en exécutant la marque associée. Dans ce cas, le menu n'apparaît pas.

Les Marking menus constituent un exemple de techniques d'interaction présentant de nombreux avantages validés expérimentalement. Pourtant, nous observons que les interfaces graphiques ont peu évolué et qu'en pratique les menus linéaires restent quasiment la seule technique utilisée. Beaudouin-Lafon identifie deux raisons principales qui freinent l'évolution des interfaces graphiques [Beaudouin-Lafon 04].

La première raison est l'insuffisance d'éléments de conception [Beaudouin-Lafon 04]. « Les concepteurs d'application ont besoin de modèles, méthodes, outils qui permettent de transférer ces techniques dans des applications commerciales ». En effet, face au foisonnement de techniques de menus, il est difficile d'identifier les avantages et les inconvénients d'une

technique, ceux-ci dépendant du contexte. Les solutions avancées sont relativement peu conceptualisées et comparées. Aussi, nous visons à fournir un espace de conception permettant de caractériser et de comprendre les propriétés interactionnelles des menus.

La seconde raison est que le modèle WIMP est tellement ancré dans nos interfaces graphiques qu'il est difficile de changer les habitudes des utilisateurs. Pour qu'une nouvelle technique soit acceptée il est nécessaire d'une part qu'elle remplisse les fonctions de la technique précédente et d'autre part que ses avantages compensent le coût d'apprentissage. C'est pourquoi, l'un de nos objectifs est de concevoir et d'évaluer de nouvelles techniques de menus capables d'être une véritable alternative aux menus linéaires.

Dans cette thèse, nous visons donc la caractérisation, la conception et l'évaluation de techniques de menus.

2 Contributions

Nous résumons les contributions principales de nos travaux selon deux axes d'étude : la caractérisation des menus existants et la conception et l'évaluation de nouvelles techniques.

2.1 Caractérisation des menus

Tout d'abord, nous proposons un espace de conception des techniques de menus appelé **MenUA (Menu Utilisabilité Applicabilité)**. Nous établissons une liste de critères utiles pour identifier finement les avantages et les inconvénients respectifs de ces techniques. Pour chaque critère, nous explicitons, décrivons et comparons des solutions que nous illustrons par les techniques existantes. Notre espace MenUA vise plusieurs objectifs :

- Dresser un état de l'art des travaux sur les menus vis-à-vis de nos critères. Cette synthèse offre une vue globale et structurée du domaine favorisant sa compréhension.
- Favoriser l'introduction des nouveaux menus dans les applications. Grâce aux critères proposés, MenUA permet aux concepteurs d'application de faire des choix informés pour choisir le menu le plus adapté à l'application considérée.
- Faciliter la conception de nouvelles techniques. En mettant en évidence les éléments moins étudiés de l'espace, MenUA permet aux concepteurs de menus d'améliorer ou de créer de nouvelles techniques. De plus, il permet de guider la conception en s'appuyant sur des critères et leurs aspects lors de la conception.
- Aider la comparaison analytique de techniques en les caractérisant par des propriétés interactionnelles élémentaires et donc plus faciles à analyser.

2.2 Conception et évaluation de menus

2.2.1 Conception de quatre nouvelles techniques de menus

En s'appuyant sur les critères de notre espace MenUA, nous définissons quatre nouvelles techniques de menus.

Les **Wave menus** améliorent la navigation dans les Marking menus hiérarchiques tout en exploitant au mieux l'espace disponible grâce à une représentation concentrique inversée.

Les **Flower menus** s'appuient sur un nouveau vocabulaire gestuel exploitant, en plus de l'orientation, la courbure et le sens. Ils permettent d'augmenter le nombre de commandes dans les Marking menus, de mettre en valeur des groupements internes et de favoriser l'apprentissage du mode expert.

Les **Leaf menus** augmentent les menus linéaires de raccourcis gestuels et facilitent la sélection de commandes sur dispositifs mobiles.

Les **Multi-Touch Menus (MTM)** exploitent les possibilités nouvelles offertes par les surfaces multi-points. Ils permettent de naviguer et sélectionner des commandes avec les cinq doigts de la main.

Ces quatre techniques de menus permettent d'illustrer l'aspect génératif de notre espace MenUA pour la conception.

2.2.2 Evaluation de menus

Nos contributions résident enfin dans les résultats expérimentaux issus des évaluations que nous avons menées.

Nous avons réalisé une expérimentation afin d'évaluer le mode novice de plusieurs Marking menus. Les résultats obtenus sont différents de ceux obtenus lors de précédentes expérimentations portant sur le mode expert. Ils soulignent l'importance de la prise en compte du mode novice lors de la conception de nouvelles techniques de menus. De plus, cette évaluation a été utilisée pour valider expérimentalement les apports des Wave menus.

La seconde expérimentation a porté sur l'apprentissage du mode expert des menus. Nous avons comparé l'apprentissage des raccourcis claviers des menus linéaires à une technique de Marking menus nommée « Polygon Menus » [Zhao 06]. Cette évaluation a également été utilisée pour valider expérimentalement les apports du Flower menus.

3 Organisation du mémoire

Ce manuscrit de thèse est organisé en deux parties, correspondant à nos deux axes d'étude.

Partie I : Caractérisation des techniques de menus

Le chapitre 2 est consacré à la terminologie adoptée et introduit les concepts utiles à l'étude des menus. Nous proposons une définition et une classification des techniques de menus et nous cernons la couverture de nos travaux. Nous identifions ensuite trois axes pertinents pour l'étude des menus : *Système, Utilisateur et Interaction*.

Le chapitre 3 est dédié à la présentation générale de notre espace de conception, noté MenUA. Nous décrivons ses fondements qui sont issus des travaux existants sur les espaces de conception, puis sa structure globale en deux facteurs, l'utilisabilité et l'applicabilité. Ces deux facteurs sont détaillés dans les deux chapitres suivants.

Le chapitre 4 a trait au facteur d'utilisabilité, le premier volet de notre espace MenUA. Nous présentons un large ensemble de solutions vis-à-vis de 3 critères d'utilisabilité (la *rapidité et la précision, l'apprentissage et la mémorisation et la satisfaction*).

Le chapitre 5 concerne le facteur d'applicabilité, le second volet de notre espace de MenUA. De manière similaire, nous décrivons un éventail de solutions au regard de 3 critères : *l'adéquation à l'application, l'adéquation à la plateforme et l'adéquation à la tâche*.

Partie II : Conception et Evaluation des techniques de menus

Le chapitre 6 est consacré à l'exploration dans les systèmes de menus en mode novice. Nous présentons les Wave Menus, une nouvelle technique de menus qui améliore le mode novice des Marking menus. Nous présentons ensuite une évaluation expérimentale comparant les performances des Wave menus avec 3 autres Marking menus pour une tâche de navigation.

Le chapitre 7 est dédié au critère *apprentissage et mémorisation* pour les techniques de menus. Nous présentons les Flower menus, une nouvelle technique de menus qui augmente le nombre de commandes disponibles des Marking menus sans dégrader l'apprentissage du mode expert. Nous validons ensuite expérimentalement les Flower menus par une évaluation qui les compare aux menus linéaires et aux Polygon menus.

Le chapitre 8 concerne les menus pour les surfaces tactiles. Nous proposons d'abord les Leaf menus adaptés aux petits écrans tactiles des dispositifs mobiles. Nous proposons ensuite les Multi-Touch Menus, qui exploitent la technologie multi-points des surfaces tactiles afin de naviguer et sélectionner des commandes avec les cinq doigts de la main.

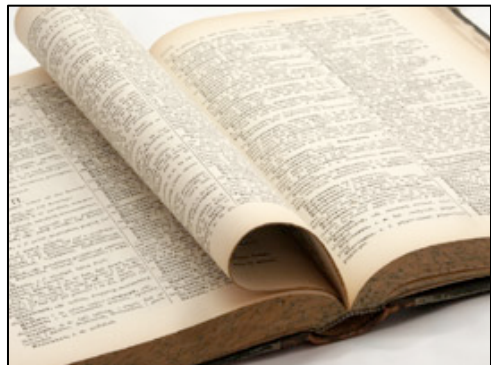
Enfin, le chapitre 9 conclut ce mémoire et présente des perspectives à ces travaux de recherche.

Caractérisation des Techniques de Menus

Partie I

Définitions, concepts et terminologies

Chapitre 2



1	MENUS	13
1.1	DEFINITIONS	13
1.2	CLASSIFICATIONS	19
2	MENU D'UN POINT DE VUE SYSTEME	21
2.1	MODE NOVICE	21
2.2	MODE EXPERT	22
2.3	MODE INTERMEDIAIRE	22
3	MENU D'UN POINT DE VUE UTILISATEUR	23
3.1	EXPERTISE DE L'UTILISATEUR	23
3.1.1	UTILISATEURS NEOPHYTES	24
3.1.2	UTILISATEURS INEXPERIMENTES	24
3.1.3	UTILISATEURS EXPERIMENTES	24
3.2	PRECISION DU BUT DE L'UTILISATEUR	24
3.2.1	ABSENCE DE BUT DE RECHERCHE	25
3.2.2	RECHERCHE D'UNE FONCTIONNALITE	25
3.2.3	RECHERCHE D'UNE COMMANDE	25
4	MENU D'UN POINT DE VUE INTERACTION	26
4.1	CHEMIN FORTEMENT DETOURNE	27
4.2	CHEMIN FAIBLEMENT DETOURNE	27
4.3	CHEMIN DROIT	28
4.4	CHEMIN DIRECT	29
5	SYNTHESE	30



Figure 1 : Illustration extraite de [Shneiderman 92]

Avant, de présenter notre espace de conception pour les techniques de menus (chapitre 3), il est nécessaire de donner une définition précise du terme "menu", d'expliquer leur fonctionnement et d'indiquer comment les utilisateurs interagissent avec. Ceci permettra de figer la terminologie adoptée et de cerner la couverture des travaux de recherche. Notre démarche, schématisée à la Figure 2, est la suivante :

La section 1 explique ce qu'est un menu en définissant et en classant les menus.

La section 2 est dédiée aux menus selon un point de vue système. Nous y introduisons la notion de mode de fonctionnement.

La section 3 est consacrée aux techniques de menus selon un point de vue utilisateur et présente deux axes : l'expertise de l'utilisateur et la précision du but de l'utilisateur. Ces deux axes nous permettent de différencier finement les cas d'utilisation des menus.

Enfin, la section 4 expose la manière dont les utilisateurs interagissent avec un menu. Nous identifions 4 types de chemins que l'utilisateur peut emprunter pour accéder à une commande dans un menu, en fonction des deux axes de la section 3. Pour chacun de ces chemins, le mode de fonctionnement utilisé (section 2) est explicité.

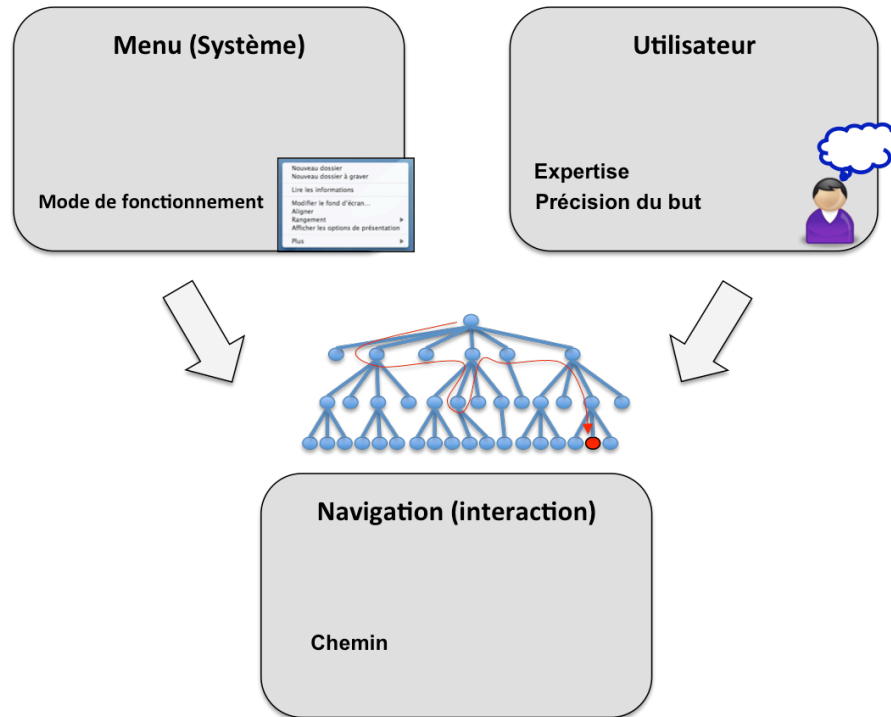


Figure 2 : Démarche d'analyse et structure du chapitre.

1 Menus

Dans cette section, nous définissons un menu puis nous classons les menus existants.

1.1 Définitions

Nous pouvons trouver dans la littérature des termes tels que « menus », « techniques de menus », « techniques à base de menus », « systèmes de menus », « menus hiérarchiques », « menus contextuels », « système de commandes »¹. Cependant, ces termes sont rarement définis avec précision. Ce constat nous incite à proposer une définition des menus, qui sera utilisée dans la suite de ce manuscrit.

Notre définition s'appuie sur le modèle d'interaction instrumentale [Beaudouin-Lafon 00] (présenté ci-après) et de deux définitions existantes de la littérature.

Modèle d'interaction instrumentale

Le modèle de l'interaction instrumentale [Beaudouin-Lafon 97, Beaudouin-Lafon 00] propose d'étudier une interface en termes *d'objets du domaine* et *d'instruments d'interaction* pour agir sur les objets du domaine (Figure 3).

Les objets du domaine sont les objets d'intérêt de l'utilisateur. Par exemple, dans un éditeur de texte, l'attention de l'utilisateur porte sur le document textuel.

L'instrument d'interaction est un médiateur entre les actions de l'utilisateur et les objets du domaine. Il transforme les actions de l'utilisateur en commandes qui agissent sur les objets du domaine. Il réagit aux actions de l'utilisateur et fournit un retour d'information sur l'état de l'objet du domaine qu'il contrôle. La Figure 3 illustre un instrument de navigation : la barre de défilement. L'instrument est composé d'une souris physique et d'une barre de défilement à l'écran. L'utilisateur interagit avec l'instrument à l'aide d'une action directe et reçoit un retour d'information direct de la souris (par le toucher) et de la barre de défilement (mise en surbrillance). La barre de défilement transforme alors les actions de l'utilisateur en des commandes de défilement pour le document. Ce dernier fournit deux types de réponse : vers la barre de défilement (mise à jour de la position de la poignée) et vers l'utilisateur (défilement du contenu du document).

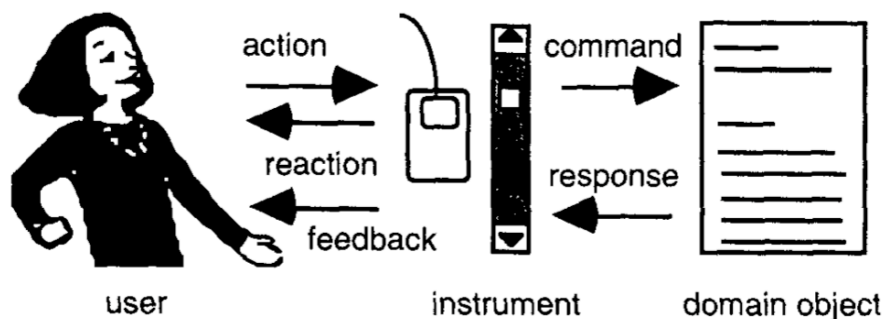


Figure 3 : Modèle de l'interaction instrumentale (illustration extraite de [Beaudouin-Lafon 00]).

¹ En anglais, les termes correspondants sont : menus, menu techniques, menu systems, menu-driven systems, hierarchical menus, multi-level menus, context menus, pop-up menus.

Il est courant que l'attention se focalise temporairement sur l'instrument lui-même. Par exemple, un crayon est un instrument pour écrire, mais c'est aussi l'objet du domaine sur lequel va agir un taille-crayon. Dans [Beaudouin-Lafon 00], un méta-instrument est défini comme un instrument qui agit sur un autre instrument.

Les menus sont donc des méta-instruments dans le modèle de l'interaction instrumentale. Ils sont utilisés pour sélectionner des commandes et des outils, c'est-à-dire pour activer des instruments.

Bien que nous nous appuyions sur la notion de méta-instrument dans notre définition, elle n'est pas suffisante, d'un point de vue système, pour différencier un menu de certains autres interacteurs comme par exemple une palette (qui est également un méta-instrument). De plus, elle ne met pas en évidence certaines caractéristiques importantes d'un menu vis-à-vis de l'utilisateur comme le fait qu'un menu offre un ensemble d'options à l'utilisateur. Pour ces raisons, nous présentons maintenant deux définitions qui adoptent un point de vue utilisateur.

Définitions existantes

Plusieurs définitions existantes [Lee 91, Norman 91, Jackoby 92, Barfield 93, Marcus 94, Helander 97, Shneiderman 92, Foley 90, ISO 9241-14] sont trop générales pour permettre de distinguer clairement les menus des autres interacteurs. Ainsi, la norme ISO 9241-14 [ISO 9241-14] définit un menu comme un « ensemble d'options que l'on peut sélectionner ». Cette définition ne permet pas de différencier les menus, des palettes, des boîtes de dialogue ou encore des boîtes à options. De notre revue de la littérature, nous avons retenu deux définitions qui nous semblent plus précises [Norman 91, Helander 97] :

Menu selection is a mechanism for users to indicate their choices. The characteristics of menu selection are that a) the interaction is, in part, guided by the computer; b) the user does not have to recall commands from memory, and (c) user response input is generally straight forward.

Psychology of Menu selection, 1991 [Norman 91]

Menus are as a set of options, displayed on the screen where the selection and execution of one (or more) of the options results in a change in the state of the interface

Handbook of Human Computer Interaction, 1997 [Helander 97]

Ces définitions soulignent les notions de choix et de reconnaissance. L'utilisateur peut choisir une commande sans être forcé de se rappeler son nom comme c'est le cas avec les langages de commandes. Notre définition d'un menu s'articule selon cinq caractéristiques dont les deux caractéristiques ci-dessus que sont le choix et la reconnaissance. Nous organisons ces cinq caractéristiques en deux classes : les caractéristiques correspondant à un *point de vue utilisateur* et à un *point de vue système*.

Caractéristiques du point de vue de l'utilisateur

D'un point de vue utilisateur, les menus possèdent trois caractéristiques qui nous semblent essentielles. Ils 1) offrent un choix à l'utilisateur, 2) réduisent l'effort mental, et 3) organisent les éléments.

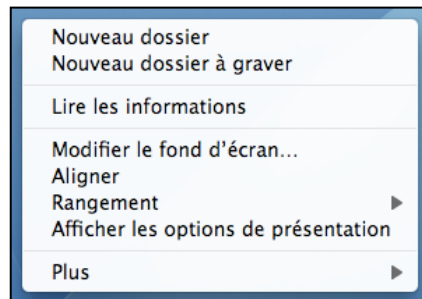


Figure 4 : Menu dont les éléments sont discrets (commandes).

1) Les menus offrent aux utilisateurs la possibilité de choisir un élément dans un ensemble borné. Cet élément peut être une commande, un choix ou un état associé à une commande. Il peut être discret (une liste de commandes, Figure 4) ou continu (le niveau de volume dans Mac OS X, Figure 5).

Cette caractéristique est rattachée aux citations suivantes : « Un ensemble d'options que l'on peut sélectionner² » [ISO 9241-14, Foley 90], « Les menus permettent aux utilisateurs de naviguer et sélectionner un item dans une liste de choix ou de commandes de l'application³ » [Marcus 94].



Figure 5 : Menu dont les éléments sont continus (le niveau sonore) dans un ensemble borné.

2) Les menus réduisent l'effort mental de l'utilisateur. Ils exploitent la reconnaissance plutôt que le rappel. Ils sont souvent opposés aux langages de commande (Figure 6).

Cette caractéristique est rattachée aux citations suivantes : « Les menus proposent une interface facile à utiliser pour sélectionner des commandes plutôt que de devoir se les rappeler. Ils facilitent la prise de décision et réduisent la charge cognitive de l'utilisateur⁴ » [Marcus 94]. « Les nouveaux utilisateurs trouvent les menus attractifs parce que toutes les options sont visibles et il n'y a pas besoin de les mémoriser ⁵ » [Barfield 93], « les menus exploitent la reconnaissance plutôt que le rappel...⁶ » [Lee 91]. « Les menus sont attractifs parce qu'ils éliminent l'entraînement et la mémorisation de séquences de commandes complexes » [Shneiderman 92].

² Set of selectable options [ISO 924-14]. Menu selection is one of the richest technique for selecting from a relatively fixed sized choice set [Foley 90].

³ Menu ... allow the user to browse and select an item from a list of choices or commands that the application provides [Marcus 94].

⁴ Menu provide an easy-to-use visual interface ... to select commands rather than having to recall the commands ... from memory. This makes the decision-making process easier and reduces the user's information load [Marcus 94].

⁵ New users find menus attractive because all the options are visible and there is no need to remember commands [Barfield 93]

⁶ Menu exploit recognition rather than recall [Lee 91].

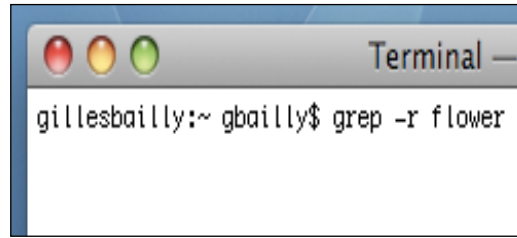


Figure 6 : Ligne de commande qui force les utilisateurs à se rappeler le nom et la syntaxe de la commande.

3) Les menus présentent les éléments selon une organisation structurée sémantiquement et spatialement. Les commandes sont en effet généralement arrangées dans des groupements hiérarchiques ou logiques (entre deux séparateurs) illustrés Figure 4. Elles peuvent également être ordonnées alphabétiquement, numériquement, sémantiquement ou en fonction de la fréquence d'accès. Spatialement, les menus ont une structure géométrique définie (linéaire, circulaire, etc) pour organiser les items [Jackoby 92, Dachselt 07]. Cette notion d'organisation est souvent absente de la littérature lorsqu'il s'agit d'explicitier les caractéristiques internes des menus. Pourtant, elle nous semble essentielle car les éléments ne sont jamais placés aléatoirement.

Nous considérons également 2 caractéristiques des menus d'un point de vue *système*.

Point du point de vue système

D'un point de vue système, les deux principales caractéristiques d'un menu sont qu'il est transitoire et quasimodal.

4) Les menus sont transitoires (*transients* [Jackobsen 07]). Ils sont affichés seulement temporairement : ils n'occupent pas une zone de l'écran de manière permanente. Ils apparaissent sur demande de l'utilisateur et disparaissent après l'activation d'une commande ou une annulation. Les menus ne sont pas les principaux objets d'intérêt de l'utilisateur et ne doivent donc pas masquer sa zone de travail.

5) Les menus sont *quasimodaux*. Notre définition s'appuie sur celle de Raskin [Raskin 00]. Les quasimodes sont « des modes qui sont conservés seulement durant l'interaction avec l'utilisateur ». « Un quasimode a les avantages d'un mode mais ne force pas l'utilisateur à se rappeler l'état actuel de l'application ». Nous distinguons deux degrés de quasimodalité (en latin quasi signifie presque, à un certain degré) : fort et faible. Pour ceci nous nous appuyons sur la nature du retour d'information utilisé [Sellen 90].

Un quasimode fort, également appelé mode à ressort (« *spring-loaded mode* ») est un mode qui n'existe que durant une action consciente et physique de l'utilisateur. Le retour d'information associé aux quasimodes est kinesthésique, c'est-à-dire que le retour d'information repose sur une tension musculaire. Il a la particularité de maintenir l'attention de l'utilisateur et de prévenir les erreurs de modes [Sellen 90]. L'exemple bien connu d'un quasimode fort est l'utilisation de la touche Shift qui est valable uniquement lorsque l'utilisateur a le doigt enfoncé sur cette touche. Les premiers menus, les *spring-loaded menus*, et la plupart des menus à base de gestes sont fortement quasimodaux car ils nécessitent que l'utilisateur garde le bouton de souris enfoncée durant toute l'interaction. Un inconvénient du maintien d'une touche enfoncée est que l'utilisateur peut éprouver de la fatigue (en particulier pour les personnes âgées).

Un quasimode faible ne nécessite pas un retour d'information kinesthésique durant toute la durée du mode. Cependant, l'utilisateur sait de manière sûre et évidente dans quel mode il est grâce à un retour d'information visuel explicite par exemple. De plus, le mode est désactivé automatiquement à la fin de l'interaction. La plupart des menus linéaires sont utilisés avec une

faible quasimodalité⁷ : l'interaction est clairement délimitée d'un côté par l'ouverture du menu et de l'autre côté par l'activation d'une commande (ou d'une annulation).

Un point important est que tous les événements du système sont capturés (« grabbed ») par le menu lorsqu'il est ouvert. Quel que soit l'objet graphique qui se trouve en dessous du pointeur, les événements sont redirigés vers le menu.

Définitions

Nous définissons un *menu* comme un méta-instrument d'interaction [Beaudouin-Lafon 00] qui vérifie les cinq caractéristiques ci-dessus.

Un *système de menus* est un ensemble de menus liés entre eux. Par exemple, les menus hiérarchiques constituent un système de menus.

Une *technique de menus* qualifie la "technique d'interaction" [Appert 04] associée à un menu ou à un système de menus (« look and feel »).

Le *menu courant* est le menu sur lequel interagit l'utilisateur à un moment donné dans un système de menus (Figure 7).

Un *sous-menu* est un menu que l'on peut accéder à partir d'un item du menu courant (Figure 7). Le nom de l'item est le titre du sous-menu.

Un *super-menu* ou un *menu parent* est un des menus qui a permis d'accéder au menu courant (Figure 7).

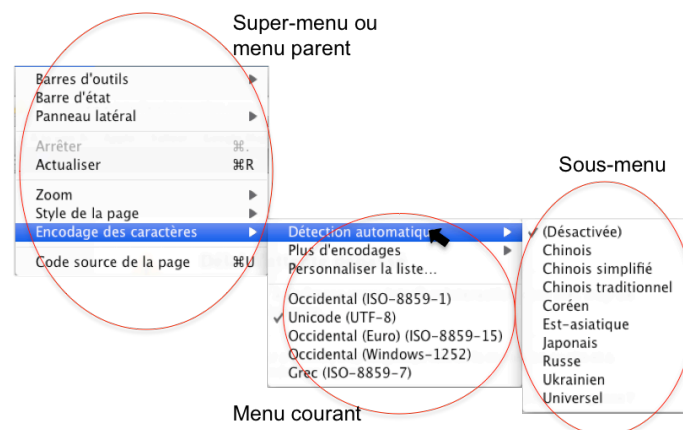


Figure 7 : Relations entre le menu courant, un super menu (ou menu parent) et un sous-menu.



Nous retenons qu'un menu est un méta-instrument d'interaction transitoire et quasimodal qui réduit l'effort mental de l'utilisateur en lui proposant un choix organisé.

Un système de menus est un ensemble de menus liés entre eux. Une technique de menus qualifie la technique d'interaction associée à un menu ou à un système de menus.

⁷ La plupart des menus linéaires peuvent être utilisés soit dans un quasimode fort en gardant le bouton de la souris enfoncé (dragging state) soit dans un quasimode faible en gardant le bouton relevé et en cliquant sur les items (tracking state).

Les non-menus

Certains interacteurs parfois qualifiés de menus ne vérifient pas les caractéristiques de notre définition. Ainsi, les palettes sont parfois confondues avec les menus car permettant elles-aussi de choisir des commandes (caractéristique 1). Cependant, les palettes ne sont pas transitoires (caractéristique 4) car elles sont affichées en permanence à l'écran. Les Tracking menus [Fitzmaurice 03] et leur amélioration les Pie cursor [Fitzmaurice 08], illustrés Figure 8, sont des techniques pour la sélection de commandes qui restent affichées en permanence à proximité du curseur. Ces deux techniques dénotées comme des menus par leurs auteurs sont en fait des palettes.

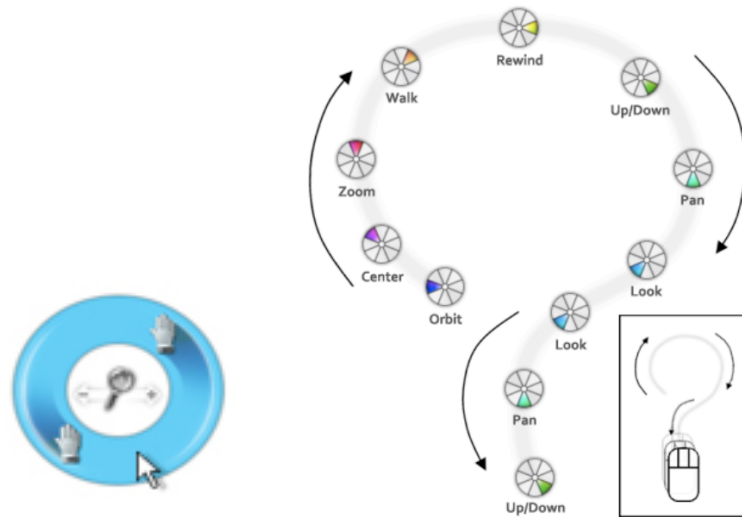


Figure 8 : A gauche les Tracking menus [Fitzmaurice 03], à droite les Pie Cursors [Fitzmaurice 08] (illustrations extraites de [Fitzmaurice 08]).

Similairement, les boîtes de dialogue ne sont pas des menus car elles sont généralement modales (et non pas quasimodales), ne proposent pas réellement de choix structuré et nécessitent une étape supplémentaire de validation.



Figure 9 Combobox.

Enfin, certains interacteurs sont assimilés à des menus bien qu'ils soient plutôt une combinaison d'interacteurs. Par exemple, les menus d'options (« Option menus ») ou les « Combobox » sont des interacteurs qui affichent en permanence l'élément sélectionné dans un interacteur (label, champ textuel, bouton) qui sert d'« ancre » pour ouvrir un menu (Figure 9). En d'autres termes, un menu d'options ou une combobox est une combinaison de deux composants : une ancre et un menu. Similairement, la barre de menu (Figure 10-a) n'est pas un menu car elle est toujours visible, mais une ancre qui sert à ouvrir un ou plusieurs menus.

1.2 Classifications

Tandis qu'il n'existe pas de définition consensuelle de la notion de menu, la classification ci-dessous en quatre catégories est souvent utilisée [Marcus 94].

Classification existante

Traditionnellement, les menus se répartissent en quatre types :

- *Les pull-down menus* ou menus déroulants, qui sont activés à partir de la barre de menus (Figure 10-a).
- *Les cascading menus*, aussi appelés pull-right menus, sont une sorte de pull-down menus qui sont dépliés généralement à droite d'un item d'un menu parent (Figure 10-b).
- *Les pop-up menus*, aussi appelés menus contextuels (ou plus rarement « Walking menus » [Walker 90]) sont activés lorsque l'utilisateur clique sur un objet d'intérêt (généralement avec le bouton droit de la souris) (Figure 10-c).
- *Les Tear-off menus* peuvent être détachés de la barre des menus et rester affichés jusqu'à ce que l'utilisateur les ferme. Lorsque l'utilisateur appuie sur un bouton, généralement représenté par une ligne discontinue, le menu se transforme en une palette qui reste affichée en permanence à l'écran (Figure 10-d).

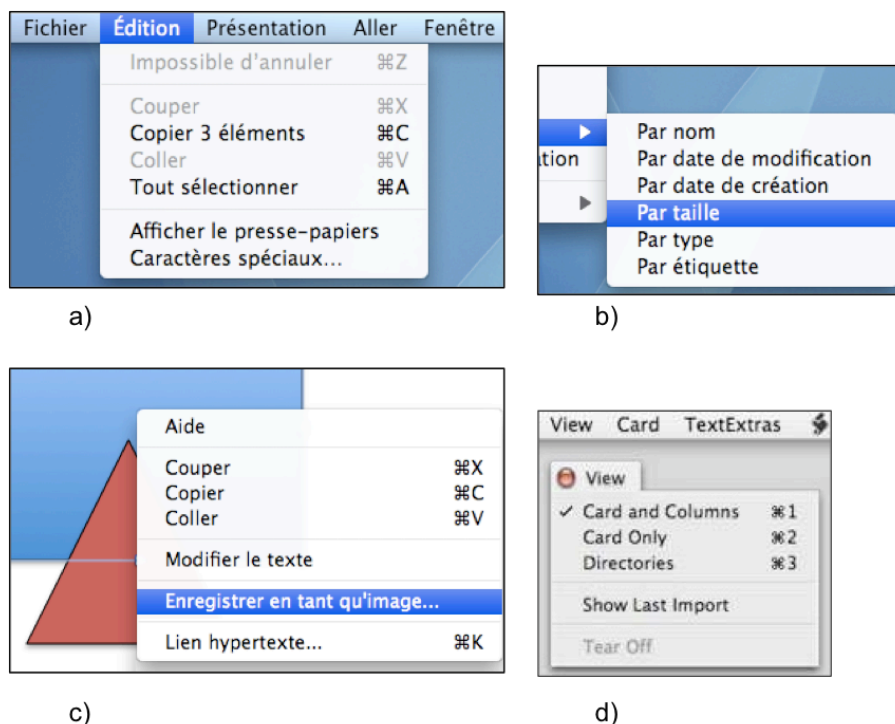


Figure 10 : 4 types de menus. (a) Les « Pull-down menus », (b) Les « Cascading menus », (c) « Pop-up menus », (d) « Tear-off menus ».

Nous identifions trois inconvénients à cette classification des menus :

- Les trois premiers types de menus sont très similaires. La seule différence est la nature des interacteurs parents : la barre des menus pour les « pull-down menus », un autre menu pour les « Cascading menus » ou un autre interacteur pour les « pop-up menus ». Ces trois menus ne sont donc pas différenciés par leurs caractéristiques intrinsèques, mais par la nature des interacteurs parents.

- Les « Tear-off menus » sont particuliers de par leur nature hybride : selon le mode d'utilisation, ils se comportent comme des menus ou des palettes. De plus ils sont rarement utilisés aujourd'hui.
- Enfin, cette classification ne prend pas en compte les menus circulaires comme les Pie menus [Callahan 88] ou les Marking menus [Kurtenbach 91]. Aussi, pour nos travaux, nous considérons la classification en deux catégories suivantes.

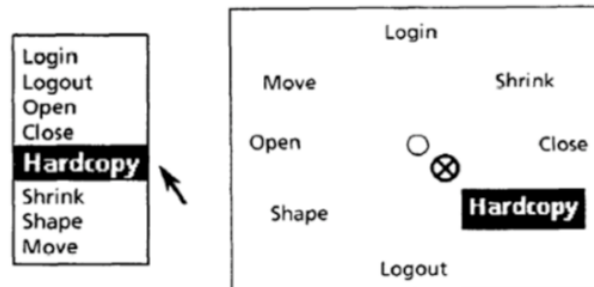


Figure 11 : À gauche un menu linéaire et à droite un menu circulaire (illustration extraite de [Callahan 88]).

Classification adoptée

Comme dans [Zhao 08], nous distinguons les menus linéaires (quelque soit la nature de l'interacteur parent) des menus circulaires (dont la première version est apparue en 1969 [Wiseman 69, Forsey 84] jusqu'aux versions les plus connues : les Pie menus [Callahan 88] illustrés Figure 11 et les Marking menus (Figure 13) [Kurtenbach 91]). Contrairement, à la classification précédente, nous ne différencions pas les différents types de menus linéaires qui sont très similaires. Deux considérations justifient la classification : *l'organisation spatiale* et *le mode de fonctionnement*.

- L'organisation spatiale est différente pour ces deux types de menus. Les items sont organisés verticalement dans les menus linéaires. C'est la distance qui permet de différencier les items. Au contraire, dans les menus circulaires, les items sont organisés autour du point d'activation et c'est l'orientation qui permet de différencier les items. Toutes les commandes sont à égale distance du centre et peuvent être associées à une direction.
- Une différence peut être plus essentielle entre ces deux catégories est leur mode de fonctionnement en mode expert. Celui des menus linéaires est basé sur des raccourcis clavier alors que celui des menus circulaires est basé sur l'interaction gestuelle. Nous détaillerons les modes de fonctionnement des menus dans la section suivante.

Au delà des considérations techniques, cette classification se justifie aussi par un relatif cloisonnement des travaux de recherche :

- Les auteurs de menus conçoivent des menus dans une même catégorie.
- Les améliorations portées dans l'une des catégories sont rarement reportées dans l'autre catégorie.
- Les évaluations expérimentales comparent généralement des menus appartenant seulement à une même catégorie.
- Les types d'évaluation sont différents (chapitres 6 et 7).

Ce cloisonnement constaté des travaux souligne le réel besoin de disposer d'un cadre fédérateur rassemblant les *solutions* proposées dans ces deux catégories. Ceci afin de faciliter leur comparaison et mettre en évidence leurs similitudes. Ce cadre fédérateur, *MenUA*, sera présenté dans le chapitre suivant.



Nous retenons, dans ce manuscrit de thèse, une classification des menus en deux catégories : les menus linéaires et les menus circulaires.

Cette classification repose à la fois sur la différence de leur représentation graphique mais aussi de leur mode de fonctionnement.

Nous venons de définir les menus et de les classer en deux catégories principales. Nous détaillons maintenant leurs modes de fonctionnement.

2 Menu d'un point de vue système

La plupart des menus ont deux modes de fonctionnement : le mode novice qui est le plus souvent utilisé et le mode expert pour les utilisateurs expérimentés. Certains menus ont également un mode intermédiaire.

2.1 Mode novice

Le mode novice est basé sur la reconnaissance des commandes [Lee 91]. En explorant le menu, l'utilisateur reconnaît la commande désirée. Il navigue dans le menu généralement avec un dispositif de pointage comme la souris ou le stylet. Certains menus permettent également une interaction au clavier à l'aide des *touches directionnelles* ou des *touches mnémoniques*. Dans le premier cas, la navigation est séquentielle et impose de traverser plusieurs items avant d'accéder à l'item désiré. Dans le second cas, la navigation est directe, mais nécessite d'identifier la lettre servant de mnémorique et d'activer la touche correspondante (Figure 12). De plus, il y a des risques de collisions du fait du nombre limité de mnémoniques.

Concernant les touches mnémoniques, nous notons des différences au niveau de l'interaction entre les systèmes d'exploitation et les boîtes à outils graphiques⁸.

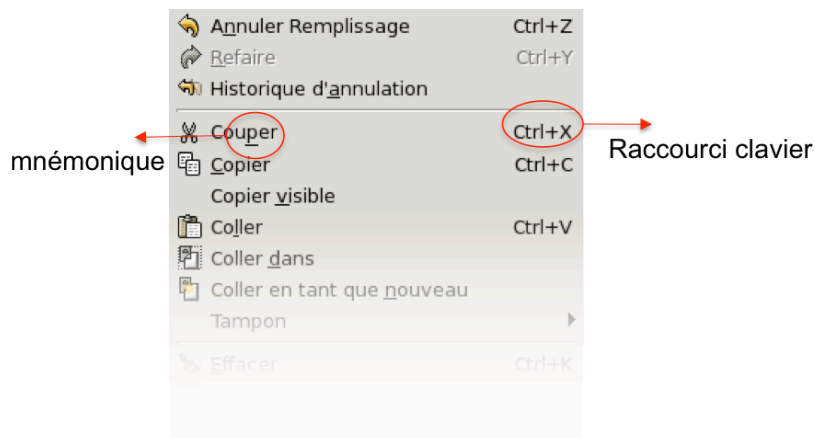


Figure 12 : Lettre servant de mnémorique (la lettre « p ») et raccourci clavier (Ctrl+X) pour la commande « Couper » du menu « Edition ». La mnémorique peut être utilisée uniquement lorsque le menu est ouvert en appuyant sur la lettre correspondante. Au contraire, le raccourci clavier peut être utilisé même lorsque le menu est fermé. Il est unique dans tout le système de menus.

⁸ Par exemple, sous Windows, la lettre servant de mnémorique est soulignée. Sous Macintosh, c'est toujours la première lettre de la commande. En cas de collision (deux commandes qui ont la même première lettre), la répétition de la touche permet parfois de les différencier.

2.2 Mode expert

Comme son nom l'indique, le mode de fonctionnement expert est utilisé par des utilisateurs plus expérimentés. Il a l'avantage d'une part d'être plus rapide et d'autre part de laisser l'utilisateur concentré sur sa tâche principale en n'affichant pas le menu dans sa zone de travail. En contrepartie, il est nécessaire que l'utilisateur apprenne le mode expert et donc qu'il mémorise comment exécuter la commande.

Le mode expert n'exploite pas la reconnaissance mais le rappel. Ceci peut sembler en contradiction avec la seconde caractéristique de notre définition (section 1.1). Néanmoins, il convient de noter que le mode expert est une alternative redondante [Jackoby 92], efficace, et répandue des menus pour favoriser la sélection de commandes.

Nous trouvons essentiellement deux types de techniques pour le mode expert : les *raccourcis claviers* et *l'interaction gestuelle*. Les raccourcis claviers sont principalement utilisés dans les menus linéaires (Figure 12). Ils consistent à associer une commande à une ou plusieurs touches du clavier. Dans les applications qui nécessitent l'entrée de texte, les raccourcis claviers combinent une touche alpha-numérique et un modificateur (Ctrl, Alt, Shift, CMD). Les raccourcis claviers sont généralement affichés à droite de la commande afin de permettre à l'utilisateur de les apprendre en utilisant le mode novice.

L'autre technique pour le mode expert repose sur l'interaction gestuelle introduite avec les Marking menus [Kurtenbach 91]. Les Marking menus, illustrés Figure 13, sont des menus circulaires qui ont deux modes de fonctionnement. Si l'utilisateur attend un certain délai (approximativement 300ms), le menu apparaît à l'écran. C'est le mode novice. L'utilisateur sélectionne une commande en déplaçant le pointeur dans la direction de l'item associé. Si l'utilisateur n'attend pas le délai, le menu passe en mode expert. Il n'apparaît pas à l'écran et l'utilisateur doit reproduire le geste qu'il a appris en mode novice. L'interaction devient alors purement gestuelle. Une propriété essentielle de cette technique est que ces gestes sont appris implicitement dans la mesure où l'utilisateur fait les mêmes gestes en mode novice. Au contraire, dans les menus linéaires, les utilisateurs sont contraints d'apprendre deux styles d'interaction différents. Nous revenons sur ce point dans la section 2.3 du chapitre 4.

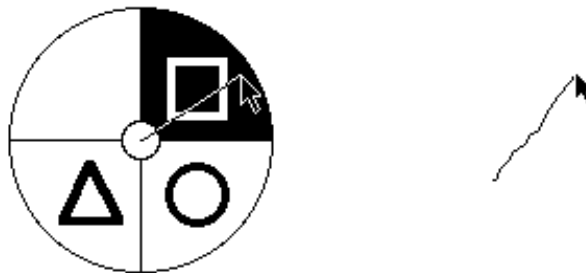


Figure 13 : Marking menus classiques [Kurtenbach 91]. A gauche, le mode novice. Après 300ms, le menu s'affiche et l'utilisateur sélectionne une commande comme dans les menus linéaires. Si l'utilisateur n'attend pas ce délai, le menu est en mode expert (à droite). Le menu n'est pas affiché et le curseur laisse une trace dont l'orientation est interprétée par le système (Illustration extraite de [Kurtenbach 91]).

2.3 Mode intermédiaire

Kurtenbach [Kurtenbach 93b] introduit un troisième mode de fonctionnement pour les Marking menus : le mode intermédiaire⁹. Ce mode est hybride entre le mode novice et le mode expert. Il intervient lorsque l'utilisateur commence à tracer une marque en mode expert puis souhaite que le système lui confirme que cette marque correspond bien à la commande désirée avant de l'activer en relâchant le bouton. En attendant 300ms, le menu apparaît et il peut finir la

⁹ Le mode intermédiaire est également appelé : mode hybride (« combined mode ») ou mode de confirmation de la marque (« mark confirmation »).

sélection de la commande en mode novice. L'utilisateur utilise donc les deux modes de fonctionnement pour la sélection d'une commande. Il commence en mode expert, puis finit en mode novice. Ce mode hybride n'est pas possible avec les menus linéaires, car dès que l'utilisateur a appuyé sur un raccourci clavier, il ne peut plus revenir en mode novice.



Nous retenons que la plupart des menus ont deux modes de fonctionnement : le mode novice et le mode expert. Certains ont un mode supplémentaire : le mode intermédiaire.

Nous avons présenté un premier axe pour l'étude des menus : le mode de fonctionnement. Cet axe est centré système car il considère le fonctionnement du menu. Aussi, dans la section suivante, nous adoptons un point de vue complémentaire centré sur l'utilisateur d'un menu.

3 Menu d'un point de vue utilisateur

Il existe de nombreux modèles de l'utilisateur [Card 83, Rasmussen 83]. Nous étudions ici les modèles permettant de mieux comprendre comment un utilisateur appréhende une technique de menus et la sélection d'une commande. Pour cela, nous considérons deux axes d'étude : *l'expertise de l'utilisateur* et *la précision du but de l'utilisateur*.

3.1 Expertise de l'utilisateur

L'expertise de l'utilisateur peut être mise en relation avec « le profil de l'utilisateur » défini par Shneiderman [Shneiderman 92] pour un système informatique en général ou « l'expérience de l'utilisateur » [Carmel 92] pour les hypermédias. L'expertise se concentre sur les connaissances globales de l'utilisateur sur le menu. Dans notre cas, l'expertise dépend principalement de la fréquence d'utilisation. Un utilisateur qui manipule le menu d'une application depuis longtemps sera plus expérimenté qu'un utilisateur qui vient de découvrir l'application. L'expertise peut être décomposée en deux dimensions orthogonales : la capacité de l'utilisateur à comprendre le fonctionnement du menu et celle à comprendre et assimiler son organisation. Un utilisateur peut très bien savoir utiliser une technique de menus sans connaître l'organisation des commandes : par exemple, lorsqu'il découvre une nouvelle application avec la même technique de menus. Réciproquement, un utilisateur peut connaître l'organisation des commandes sans savoir utiliser la technique de menus. Par exemple, un logiciel peut proposer deux techniques de menus équivalentes pour accéder à la hiérarchie de commandes. En pratique, le second exemple est rare. Pour simplifier, nous considérons que l'utilisateur apprend le fonctionnement d'un menu avant d'apprendre son organisation. Selon le modèle de Rasmussen [Rasmussen 83], nous distinguons trois niveaux d'expertise que nous appliquons au cas des menus :

- les utilisateurs néophytes,
- les utilisateurs inexpérimentés,
- les utilisateurs expérimentés.

3.1.1 Utilisateurs néophytes

Les utilisateurs néophytes ne connaissent pas la technique d'interaction considérée : ils ne l'ont jamais utilisée auparavant et doivent pouvoir comprendre son fonctionnement sans aide. Il ne suffit pas de concevoir des techniques efficaces, il est nécessaire que la prise en main soit simple sinon les utilisateurs néophytes risquent de se décourager et d'abandonner avant de pouvoir tirer profit de la nouvelle technique.

3.1.2 Utilisateurs inexpérimentés

Les utilisateurs inexpérimentés savent comment la technique fonctionne, mais ne connaissent pas bien l'organisation des commandes. Ce sont généralement des utilisateurs occasionnels qui n'utilisent qu'un petit nombre de commandes (les plus fréquentes).

3.1.3 Utilisateurs expérimentés

Les utilisateurs expérimentés sont à l'aise avec la technique d'interaction. Ce sont souvent des professionnels qui utilisent très fréquemment un logiciel. Avec l'habitude, ils ont une vue globale de la hiérarchie. Même s'ils ne connaissent pas la position d'une nouvelle commande, leur expérience leur permet d'éliminer les cas connus et de la trouver rapidement.

Les utilisateurs expérimentés s'adaptent facilement à de nouvelles applications dans leur domaine d'expertise grâce au transfert de savoir. Par exemple, un utilisateur expérimenté du menu d'Excel (tableur MS-Office) peut facilement s'approprier l'organisation du menu de Numbers (tableur propre au Macintosh) en identifiant les différences et les similitudes entre ces deux logiciels.

Nous soulignons le fait que nous avons utilisé volontairement les termes néophyte, inexpérimenté et expérimenté et non pas novice, intermédiaire et expert comme c'est le cas dans certains ouvrages [Rasmussen 83, Shneiderman 92]. En effet, cette dernière dénomination a déjà été utilisée pour distinguer les modes de fonctionnement (mode novice, mode intermédiaire, mode expert). L'utilisation du même vocabulaire pour deux notions très différentes a introduit des confusions dans la littérature sur les menus [Kurtenbach 93b, Zhao 04]. Par exemple, certains textes indiquent qu'un utilisateur expert utilise le mode expert. En réalité un utilisateur expérimenté utilise le mode expert seulement pour certaines commandes fréquentes. Pour le reste des commandes, il utilise le mode novice. Un utilisateur peut donc être expérimenté et cependant utiliser fréquemment le mode novice. Cette confusion vient du fait qu'il est important de distinguer les connaissances globales de l'utilisateur sur le menu (que nous dénotons expertise) des connaissances de l'utilisateur pour une commande particulière lesquelles sont liées à la notion de précision du but, que nous détaillons ci-après.

3.2 Précision du but de l'utilisateur

Cette notion permet de mesurer le niveau de précision du but de l'utilisateur. Est ce que l'utilisateur sait précisément ce qu'il veut faire, ce qu'il recherche ou au contraire, est ce qu'il lui manque un certain nombre d'informations pour préciser son but ? Par exemple, dans un logiciel de vidéo à la demande, l'utilisateur peut souhaiter regarder un film, mais sans idée précise (absence de but de recherche). Il peut vouloir regarder un film dont « le bateau coule » mais dont il ne se rappelle pas le nom (définition partielle de son but). Enfin, il peut désirer regarder le film « Titanic » (définition complète de son but).

Cet exemple illustre trois niveaux de précision dans la définition du but de l'utilisateur, mais en réalité c'est un continuum. Il est important de noter que cet axe réfère uniquement à la capacité de l'utilisateur à formuler son but et non à la capacité de l'utilisateur de savoir comment atteindre son but. En d'autres termes, nous ne traitons pas les connaissances procédurales de l'utilisateur, mais ses connaissances factuelles. Pour l'exemple de vidéo à la demande, nous ne

nous intéressons pas à savoir si l'utilisateur sait comment à accéder au film Titanic, mais s'il connaît son nom. Le chemin utilisé pour atteindre son but est traité dans la section suivante.

L'axe précision du but peut être mis en relation avec l'axe « Target Orientation » [Waterworth 91, Nigay 98] pour les systèmes d'information. Dans ces modèles, seuls deux cas sont distingués : l'absence ou la présence d'un but de recherche dans l'esprit de l'utilisateur. Carmel précise le second cas en distinguant un but complètement défini d'un but partiellement défini [Carmel 92]. Similairement, pour le cas des systèmes de menus, Norman différencie la recherche d'une commande dont il connaît le nom (« known goal object ») de celle d'une fonctionnalité (« unknown goal object ») [Norman 91].

En s'appuyant sur [Norman 91, Waterworth 91, Carmel 92, Nigay 98], nous identifions trois niveaux de précision dans la formulation du but de l'utilisateur : l'absence de but de recherche, la recherche d'une fonctionnalité et la recherche d'une commande.

3.2.1 Absence de but de recherche

L'absence de but de recherche se caractérise par le fait que l'utilisateur ne recherche pas une commande en particulier. Il peut quitter le menu sans avoir sélectionné de commandes. Typiquement, lorsque l'utilisateur prend en main un nouveau logiciel, il souhaite avoir un aperçu de ses possibilités, c'est-à-dire connaître quelles sont les commandes à sa disposition. Ceci correspond à l'exemple de l'utilisateur qui ne recherche pas un film en particulier et se laisse guider par ce que lui propose le système.

3.2.2 Recherche d'une fonctionnalité

L'utilisateur souhaite modifier son objet d'intérêt. Pour cela, il doit trouver dans le système de menus une fonctionnalité qui correspond à son besoin. Il n'est pas certain qu'elle existe et ignore son libellé exact. Par exemple, l'utilisateur peut passer au dessus de l'item désiré sans le sélectionner, faute de connaître le vocabulaire du domaine. Le but de l'utilisateur est donc partiellement défini. Il recherche une fonctionnalité plutôt qu'une commande précise.

3.2.3 Recherche d'une commande

L'objectif est beaucoup plus précis. L'utilisateur recherche une commande dont il connaît le libellé exact. Il n'est pas forcé de savoir sa position dans le menu. Par exemple, après une longue période d'inactivité, l'utilisateur peut se souvenir du nom de la commande, mais pas du chemin pour y accéder.



Nous retenons deux axes pertinents pour caractériser les connaissances de l'utilisateur sur un menu en général (expertise) et sur une commande en particulier (précision du but).

L'expertise de l'utilisateur et la précision de son but permettent d'étudier un menu selon un point de vue utilisateur, tandis que les modes de fonctionnement de la section précédente adoptent un point de vue système sur les menus. Il convient donc maintenant d'étudier conjointement ces deux perspectives système et utilisateur, puisque que c'est de leur couplage qu'émerge l'essence même de l'interaction (Figure 14).

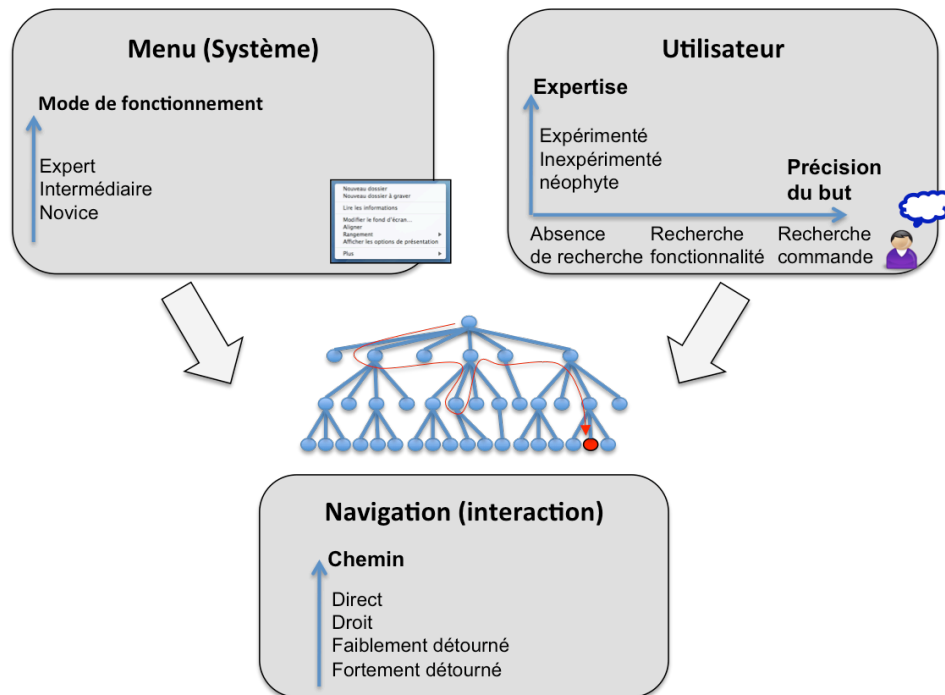


Figure 14 : Interaction avec un menu : couplage des deux points de vue, système et utilisateur..

4 Menu d'un point de vue interaction

Nous étudions ici la manière dont l'utilisateur interagit avec un système de menus, en nous focalisant sur la navigation. Pour ce faire, nous classons les différents chemins (« Route following » [Jul 97]) que peut emprunter un utilisateur pour réaliser son but, dans notre cas accéder à la commande souhaitée. Nous identifions 4 types de chemins différents, illustrés Figure 15, selon la longueur du chemin parcouru (« path length » [Catledge 95]) qui peut être fortement détourné, faiblement détourné, droit ou direct. Les trois premiers types de chemins sont inspirés de [Howes 94].

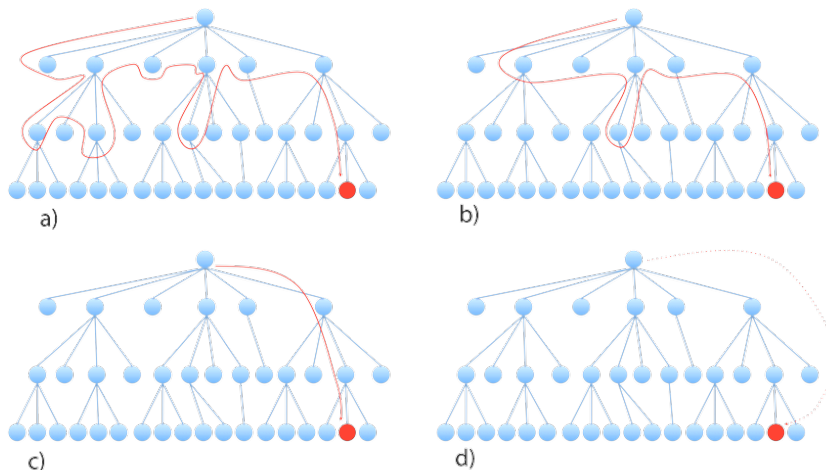


Figure 15 : Différents types de chemins pour accéder à une commande. A) chemin fortement détourné, b) chemin faiblement détourné, c) chemin droit, d) chemin direct.

4.1 Chemin fortement détourné

L'utilisateur accède à une commande de manière fortement détournée lorsqu'il est forcé d'explorer une grande partie de l'arbre de commandes pour atteindre son but. Le chemin traversé est long et caractérisé par plusieurs descentes et remontées dans la hiérarchie (Figure 15-a). Ce type de trajet est associé au mode novice (Figure 16). L'utilisateur prend ses décisions à partir des libellés des items. Plus l'utilisateur est expérimenté moins il fait des détours importants. Nous pouvons associer ce type de trajet à trois cas :

- Un utilisateur néophyte peut tester une technique de menus pour la prendre en main et visiter plusieurs branches.
- Un utilisateur inexpérimenté cherche à avoir un aperçu d'une partie ou de la totalité de la hiérarchie (découverte d'un logiciel).
- Un utilisateur inexpérimenté recherche une commande dont il ne connaît pas le chemin pour y accéder. Les titres des sous-menus ne sont pas suffisamment explicites pour lui éviter de visiter de mauvaises branches.

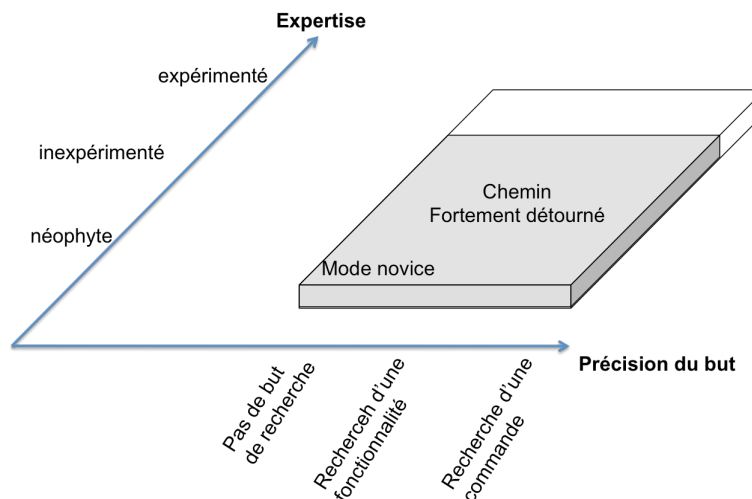


Figure 16 : Couverture et mode de fonctionnement associés à un chemin fortement détourné.

4.2 Chemin faiblement détourné

Ce type de trajet est proche du précédent. La différence est que l'utilisateur s'aventure moins profondément dans les mauvaises branches (Figure 15-b) généralement parce qu'il est plus expérimenté. Le mode de fonctionnement utilisé est toujours le mode novice (Figure 17).

L'utilisateur visite moins de mauvaises branches parce que la hiérarchie est plus petite ou parce que la qualité des groupements est meilleure. Il est important de noter que même un utilisateur très expérimenté peut faire des détours avant d'atteindre son but. C'est le cas lorsqu'il accède à des commandes peu fréquentes. Cependant, grâce à son expertise du menu, il limite le nombre de mauvaises branches visitées.

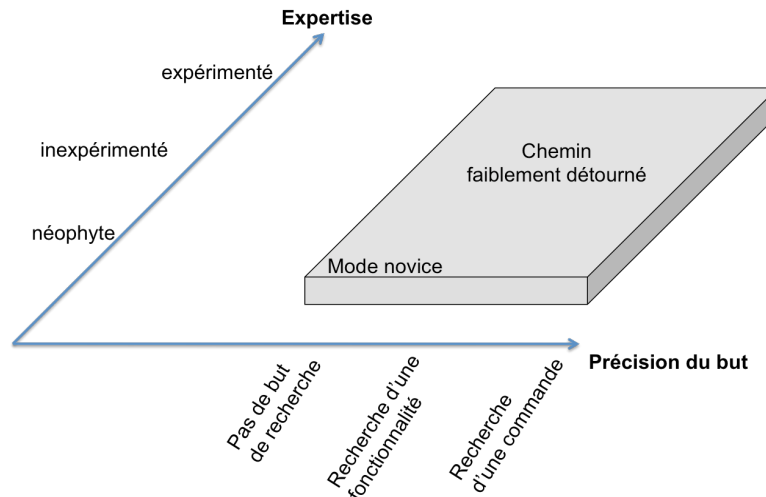


Figure 17 : Couverture et mode de fonctionnement associés à un chemin faiblement détourné.

4.3 Chemin droit

L'utilisateur suit un chemin droit pour accéder à une commande lorsqu'il ne fait aucun détour (Figure 15-c). Le mode de fonctionnement utilisé est le mode novice ou le mode intermédiaire. Il a généralement sélectionné cette commande auparavant. Il n'est pas néophyte et a un but précis (Figure 18).

Ce type de chemin peut être observé chez un utilisateur peu expérimenté. Ce dernier ne connaît pas forcément le nom de la commande, mais les titres des sous-menus le guident sans erreur jusqu'à son but. Il ne visite pas de mauvaises branches.

Ce chemin est également observé pour un utilisateur plus expérimenté qui connaît le nom de la commande et le chemin pour y accéder. Il n'a même plus besoin de lire précisément les titres des sous-menus pour atteindre son but. Cependant pour plusieurs raisons, cet utilisateur n'utilise pas le mode expert :

- Certaines catégories de personnes (personnes âgées, handicapées, etc.) ayant les connaissances adéquates pour accéder à une commande sans détour, n'oseront ou ne pourront jamais utiliser le mode expert. Elles restent donc en mode novice.
- L'utilisateur ne se sent pas encore prêt pour utiliser le mode expert et souhaite une confirmation du système avant d'activer une commande. Il utilise le mode intermédiaire.
- Il ne connaît pas le mode expert parce qu'il est difficile à apprendre. Lane montre que moins de 7% des sélections dans le logiciel Word sont réalisées en mode expert par des utilisateurs expérimentés (251 professionnels avec plus de 3 ans d'expérience) [Lane 05].
- Il ne peut pas l'utiliser parce qu'il n'existe pas. Par exemple, les dispositifs mobiles n'ont pas toujours de clavier (donc pas de raccourcis claviers).

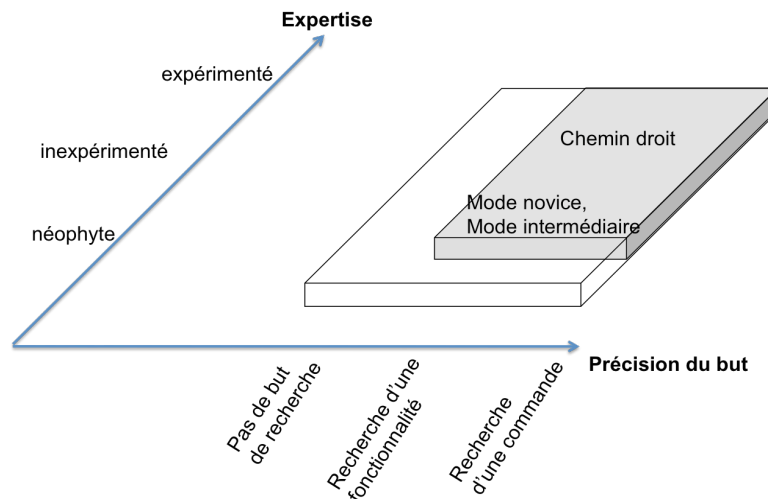


Figure 18 : Couverture et mode de fonctionnement associés au chemin droit.

4.4 Chemin direct

L'utilisateur suit un chemin direct lorsqu'il accède directement à une commande sans traverser les différents sous-menus (Figure 15-d). Par exemple, les raccourcis claviers donnent un accès direct aux commandes associées quelle que soit leur profondeur. L'interaction gestuelle associe un geste à une commande. L'accès direct est détaillé dans le chapitre 4 : il est plus rapide car il réduit le nombre d'étapes pour sélectionner une commande. Ce type de trajet est réalisable uniquement en mode expert. Les utilisateurs sont généralement expérimentés et très familiers avec la commande (Figure 19).

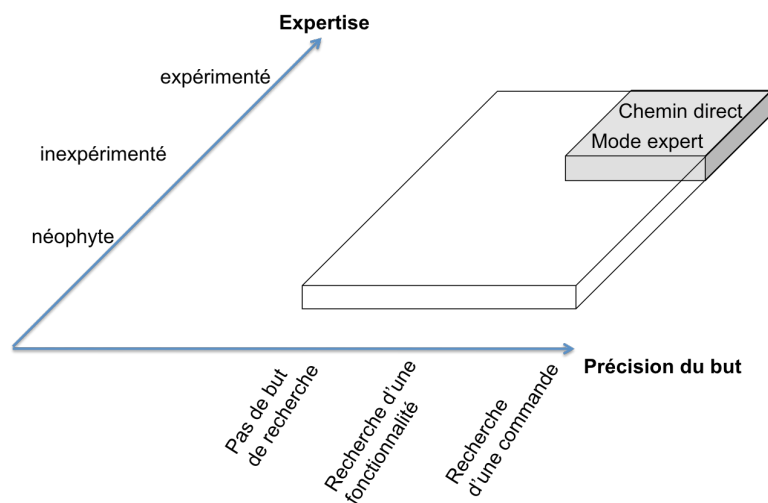


Figure 19 : Couverture et mode de fonctionnement associés au chemin direct.

5 Synthèse

Dans ce chapitre et afin de cerner le sujet de nos travaux, nous avons défini ce qu'est un menu et présenté une classification des menus en deux catégories (linéaire et circulaire).

Nous avons ensuite identifié trois axes pertinents d'étude des menus selon deux points de vue complémentaires : le système (mode de fonctionnement) et l'utilisateur (expertise et précision du but). Nous avons naturellement étudié ensuite l'interaction avec un menu en considérant ces trois axes. Nous résumons notre analyse des menus dans le tableau 1.

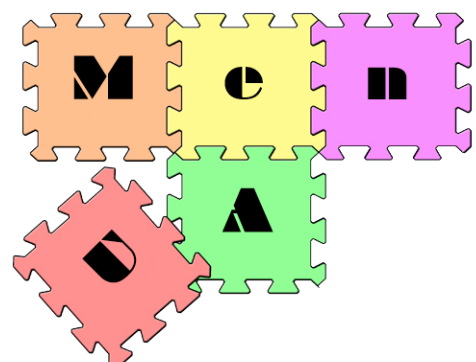
Point de vue	Axe	Définition	Valeurs
Système	Mode	Manière dont le menu fonctionne.	Novice, Intermédiaire, Expert
Utilisateur	Expertise	Connaissances globales de l'utilisateur sur le menu.	Néophyte, Inexpérimenté, Expérimenté
	Précision du but	Niveau de définition du but de l'utilisateur.	Aucun but de recherche, Recherche d'une fonctionnalité, Recherche d'une Commande
Interaction	Chemin	Chemin parcouru par l'utilisateur pour sélectionner une commande	Fortement détourné, Faiblement détourné, Droit, Direct

Tableau 1 : Les quatre axes considérés pour l'étude des techniques de menus.

La terminologie étant fixée, nous présentons au chapitre suivant notre espace de critères afin de caractériser, concevoir et évaluer des techniques de menus.

Utilisabilité et Applicabilité des menus : MenUA

Chapitre 3



1	ESPACE DE CONCEPTION	34
1.1	CRITERES DE CONCEPTION	34
1.2	MODELES DE PERFORMANCES	35
1.2.1	CIS ET SIMCIS	35
1.2.2	MODELE DE PERFORMANCE DES MENUS LINEAIRES EN MODE NOVICE	37
1.3	TAXONOMIE DES MENUS 3D	39
1.4	SYNTHESE	42
2	MENUA : STRUCTURE GLOBALE	43
2.1	DESCRIPTION	43
2.2	FACTEUR D'UTILISABILITE	44
2.3	FACTEUR D'APPLICABILITE	45
3	CONCLUSION	47

Nous observons un foisonnement des techniques de menus dans la littérature : une à deux nouvelles techniques sont proposées chaque année. Pourtant, il est relativement difficile d'identifier précisément leurs différences, similitudes, avantages et inconvénients respectifs. Face à ce constat, nous proposons MenUA, un espace de conception des techniques de menus. Il repose sur une liste de *critères* et sur une taxonomie de *solutions* basées sur ces critères. Un critère, comme la rapidité, est une mesure pour l'évaluation de la qualité d'une technique de menus. Une solution est une propriété, comme la géométrie circulaire, ou une stratégie, comme le fait d'augmenter la largeur des items de la technique de menus. Ces solutions (propriétés, stratégies) concourent à améliorer un ou plusieurs critères.

Comme tout espace de conception, MenUA vise plusieurs objectifs [Card 90, Bersen 96, Chuah 96, Dachsel 07] :

- Il dresse un état de l'art¹ des travaux sur les menus vis-à-vis des critères proposés en explicitant, décrivant et organisant les solutions existantes. L'analyse de ces dernières permettent de caractériser finement les techniques de menus de la littérature. Cette synthèse offre une vue globale et structurée du domaine favorisant sa compréhension et son assimilation.
- Il favorise la diffusion des menus. En présentant les techniques existantes, il permet aux concepteurs d'applications de choisir le menu le plus adapté à l'application considérée en se basant sur des critères. Il convient de noter que MenUA peut être utilisé par des non-spécialistes car il ne repose pas sur une représentation formelle ou semi-formelle pouvant être longue et/ou difficile à prendre en main.
- Il constitue un outil pour la conception par son aspect génératif. Tout d'abord, en identifiant des éléments moins étudiés de l'espace, il est susceptible d'avoir un effet incitatif en indiquant aux concepteurs de menus des voies intéressantes à explorer. De plus, il guide la conception en la structurant selon les critères. Enfin, il permet de détecter des faiblesses et des avantages dans les phases amont de conception
- Il facilite l'évaluation des techniques de menus. En caractérisant les techniques de menus en propriétés interactionnelles élémentaires, il aide la comparaison analytique (opposé à empirique [Senach 90]) de techniques de menus qui sont alors plus faciles à analyser. Enfin, il simplifie la mise en place d'évaluations expérimentales en identifiant précisément les critères qui sont en jeu.

Nous structurons ce chapitre en deux sections. Nous présentons d'abord les espaces de conception sur lesquels repose notre espace MenUA, puis nous décrivons la structure globale de notre espace MenUA. L'espace de conception lui-même est ensuite décrit dans les deux chapitres suivants.

¹ Le dernier état de l'art sur les menus date de 1993 [Lee 93] et n'était pas organisé en critères. Depuis, un nombre important de menus et de solutions ont été proposés

1 Espace de conception

Nous décrivons dans cette section, les espaces de conception existants et applicables au cas des techniques de menus. Un espace de conception est compris ici comme un ensemble organisé d'éléments pertinents pour la conception de techniques. Il peut prendre des formes très variées (liste de critères, modèle, taxonomie, etc.) et peut être utilisé : pour guider la conception d'une technique par un ensemble d'éléments à considérer, pour la modéliser, pour détecter des erreurs ou encore pour mesurer ses performances.

Face à l'étendue des travaux existants en termes d'espaces de conception en IHM, nous commençons par présenter les espaces de critères qui sont généraux aux techniques d'interaction et applicables aux techniques de menus. Nous présentons ensuite deux modèles permettant de mesurer l'efficacité d'une technique. Le premier est général aux techniques d'interaction, le second est spécifique aux menus linéaires. Enfin, nous introduisons une taxonomie des menus dans les environnements 3D.

1.1 Critères de conception

La littérature comprend de nombreux espaces de critères pour caractériser une application interactive [McCall 77, Abowd 92, Bastien 93, Nielsen 93, Shneiderman 04]. Issu initialement du Génie Logiciel, MacCall [McCall 77] caractérise la qualité d'un logiciel en facteurs et critères. Abowd [Abowd 92] étend cet espace de critères en affinant et structurant les critères liés à l'utilisabilité.

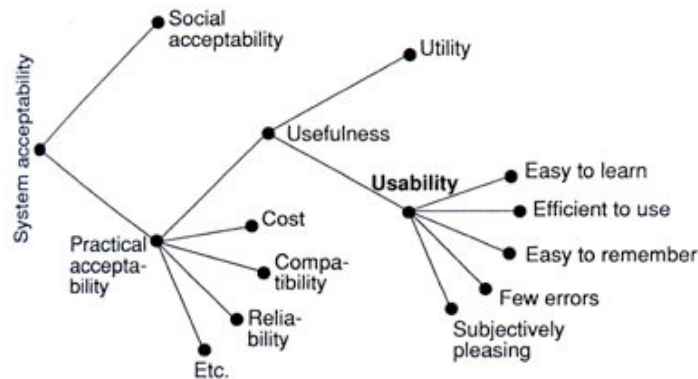


Figure 1 : Arbre de l'acceptabilité d'un système (illustration extraite de [Nielsen 93]).

En reprenant aussi une structure hiérarchique en facteurs et critères, Nielsen [Nielsen 93] évalue une application logicielle en terme d'acceptabilité. L'acceptabilité est un facteur scindé en deux dimensions (Figure 1) : l'acceptabilité sociale, s'intéressant au contexte d'utilisation du système et l'acceptabilité pratique, considérant l'intention que le système permet d'atteindre. Dans nos travaux, nous nous sommes focalisés sur la dimension « usefulness² » qui réfère à la faculté à permettre à l'utilisateur d'atteindre ses buts [Senach 90, Nielsen 93, Calvary 02]. Cette dimension se décompose en deux facteurs : le *facteur d'utilisabilité* (« usability ») et le *facteur d'utilité* (« utility ») [Nielsen 93] que nous présentons ci-après.

- **l'utilisabilité** relève de l'usage : elle traite de l'adéquation du système par rapport aux capacités cognitives, motrices et sensorielles de l'utilisateur ;
- **l'utilité**, à connotation fonctionnelle, cible une adéquation du système par rapport aux besoins de l'utilisateur.

² Usefulness n'a pas de traduction directe en français autre que « utilité » que nous utilisons déjà pour « utility ».

Shneiderman	Nielsen
Speed of performance	Efficiency
Time to learn	Learnability
Retention over time	Memorability
Rate of errors by users	Errors/Safety
Subjective satisfaction	Satisfaction

Figure 2. Correspondance entre les cinq critères d'utilisabilité de Shneiderman [Shneiderman 04] et Nielsen [Nielsen 93]. Illustration extraite de [Welie 99b].

Enfin, Nielsen [Nielsen 93] et Shneiderman [Shneiderman 04] décomposent le facteur d'utilisabilité en cinq critères que nous utilisons dans MenUA :

- la *rapidité* réfère au temps nécessaire pour accomplir une tâche ;
- l'*apprentissage* réfère au temps nécessaire pour apprendre les actions nécessaires pour accomplir une tâche ;
- la *mémorisation* réfère à la qualité du savoir retenu sur l'utilisation d'une application après une heure, une journée ou un mois ;
- la *précision* réfère au nombre d'erreurs lors de la réalisation d'une tâche ;
- la *satisfaction* réfère au plaisir de l'utilisateur à utiliser différents aspects de l'interface.

En résumé, ces canevas permettent de mesurer la qualité d'une application interactive. Ils disposent d'une large couverture grâce une grande variété de facteurs et de critères, mais proposent rarement des principes pour les mettre en œuvre. Ils permettent de détecter un certain nombre d'erreurs, mais ne permettent pas par exemple de comparer deux solutions ou de prédire le temps que mettra un utilisateur pour réaliser une tâche.

1.2 Modèles de performances

Nous présentons ici deux modèles de performances. Ces modèles de conception considèrent uniquement les critères de rapidité et de précision mais permettent de modéliser finement une technique d'interaction et de prédire ses performances.

1.2.1 CIS et SimCIS

Le modèle « Complexity of Interaction Sequence » (CIS) [Appert 04] permet de décrire la structure de techniques d'interaction et de les évaluer qualitativement. Il définit une technique d'interaction comme une séquence d'actions qui réduit progressivement l'espace de commandes à une unique commande et l'exécute. Une technique est modélisée par un graphe d'interaction (Figure 3). Les deux types d'actions élémentaires sont : les *acquisitions* et les *validations*.

- Une acquisition (représentée par un arc) identifie un sous-ensemble de l'espace de commandes courant. Il s'agit typiquement d'un déplacement du curseur vers un outil, un objet d'intérêt ou une position. Par exemple, à la Figure 3, l'utilisateur doit d'abord positionner le curseur à l'endroit approprié pour créer une forme à l'aide du menu contextuel.
- Une validation (nœud), confirme le sous-ensemble identifié par une acquisition qui devient l'espace de commandes courant. Par exemple, toujours avec le menu contextuel de la Figure 3, une fois le curseur positionné, la validation correspond à un clic de souris qui réduit l'espace de commandes aux trois commandes de création de formes.

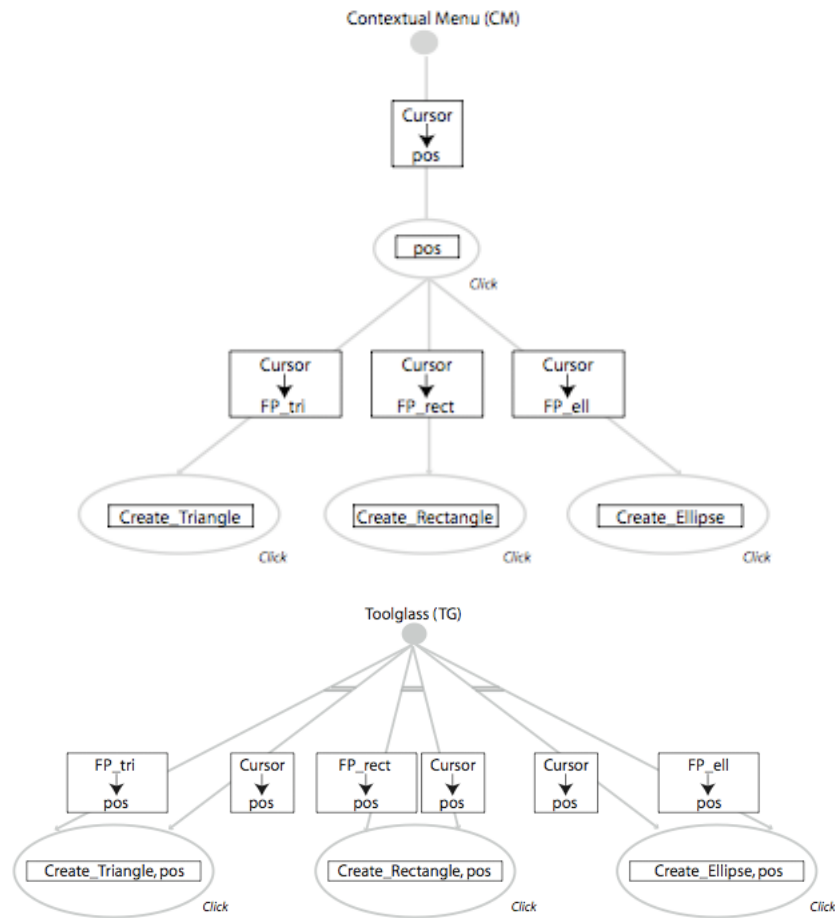


Figure 3 : Graphe d'interaction pour un menu contextuel (haut) et une Toolglass (bas) qui ont trois commandes (créer triangle, créer rectangle, créer ellipse). Illustration extraite de [Appert 04].

Les graphes CIS ont été utilisés pour concevoir de nouvelles techniques d'interaction comme OrthoZoom [Appert 06a] et ControlTree [Appert 06b] en optimisant les arcs des graphes.

Les graphes CIS mettent en évidence 4 *propriétés morphologiques* qui permettent de comparer qualitativement les techniques d'interaction :

- **Organisation temporelle** : une technique d'interaction impose une organisation séquentielle ou parallèle de ses actions. Par exemple un menu contextuel impose de choisir d'abord une position, puis l'outil. Au contraire, avec une palette, l'utilisateur choisit d'abord l'outil puis la position. Ce sont deux techniques avec une organisation séquentielle. A l'opposé, la Toolglass se caractérise par deux acquisitions parallèles (l'outil et la position) et une seule validation.
- **Persistance** : une technique peut avoir des effets de bord sur la prochaine utilisation. Par exemple, contrairement à un menu, une palette requiert une seule sélection d'outil pour la création de plusieurs formes identiques.
- **Fusion** : cette propriété réfère au contrôle de la valeur de plusieurs commandes simultanément, par exemple la largeur et le style d'une ligne dans un éditeur de dessin.
- **Développement et factorisation** : cette propriété concerne l'application de plusieurs manipulations simultanément : soit plusieurs manipulations différentes sur un objet, soit une manipulation sur plusieurs objets.

CIS est accompagné de SimCIS qui permet d'évaluer ces techniques de manière quantitative en prenant en compte le contexte. SimCIS permet, à partir d'une technique d'interaction modélisée par son graphe d'interaction et d'un contexte sous la forme d'une séquence d'interaction, de simuler l'utilisation de cette technique afin de fournir des prédictions sur son efficacité pour ce contexte donné. En d'autres termes, SimCIS permet d'évaluer quantitativement une technique d'interaction. Pour cela, il s'appuie sur trois lois empiriques.

- **La loi de Fitts** [Fitts 54] permet de prédire le temps de sélection de mouvement pour atteindre une cible de largeur W à une distance D . Elle s'exprime sous la forme $MT = a + b \cdot \log_2(1 + D/W)$ où MT est le temps de mouvement. a et b sont des constantes à déterminer empiriquement qui dépendent de différents facteurs comme le périphérique d'entrée utilisé. La loi de Fitts peut se réécrire $MT = a + b \cdot ID$, où ID est l'indice de difficulté, c'est à dire $ID = \log_2(1 + D/W)$.
- **La loi de Hick-Hyman** [Hick 52, Hyman 53] permet de prédire le temps pour choisir un élément parmi une liste de n éléments. Le temps de réaction, RT , s'exprime sous la forme $RT = a + b \cdot \log_2(n)$. Cette loi est moins fréquemment utilisée que la loi de Fitts.
- **La loi de steering** (steering law) [Accot 97] permet de prédire le temps de mouvement pour suivre un couloir rectiligne de longueur D et de largeur W . Elle s'exprime sous la forme $T = a + b \cdot D/W$.

A partir de ces trois lois empiriques, de l'état initial de l'interface, du graphe d'interaction de la technique et de la séquence d'interaction, SimCIS détermine le chemin parcouru dans le graphe, nombre d'actions et le temps total. SimCIS a été utilisé dans [Appert 04] pour comparer une palette, une toolglass et une palette bimanuelle³ pour deux contextes différents.

En résumé, le couple CIS/SimCIS considère uniquement le critère de rapidité et permet de prédire les performances de la plupart des techniques d'interaction. Les techniques basées sur l'interaction gestuelle ne sont néanmoins pas traitées. Nous retenons les 4 propriétés morphologiques (organisation temporelle, persistance, fusion et développement et factorisation) qui sont directement applicables aux techniques de menus ainsi que la prise en compte du contexte pour la conception et l'évaluation d'une technique de menus.

1.2.2 Modèle de performance des menus linéaires en mode novice

Cockburn [Cockburn 07] propose un modèle de performance pour les menus linéaires hiérarchiques. Il repose sur les mêmes trois lois empiriques que SimCIS (lois de Fitts, Hick-Hyman et de Steering) et permet d'évaluer quantitativement des techniques de menus. Le modèle se concentre uniquement sur les menus linéaires et prend en compte l'expertise⁴ de l'utilisateur.

Vue globale

Le modèle prédit le temps de sélection T_i d'un item i dans un menu à un seul niveau comme la somme du temps de recherche/décision T_{rdi} (c'est-à-dire le temps pour trouver visuellement l'item désiré) et du temps de pointage T_{pi} , (c'est-à-dire le temps nécessaire pour pointer cet item).

³ Une palette bimanuelle utilise les deux mains et deux curseurs. La main non-dominante sélectionne l'outil dans la palette avec le premier curseur et la seconde main sélectionne les objets d'intérêts avec le second curseur. Une palette bimanuelle combine donc la persistance d'une palette et l'organisation parallèle d'une toolglass.

⁴ Dans [Cockburn 07], l'expertise est entendue comme la capacité d'un utilisateur à savoir la position de la commande cible dans un menu à un niveau. L'utilisateur connaît donc déjà le nom de la commande. Dans tous les cas, l'utilisateur utilise le mode novice du menu.

$$T_i = T_{rdi} + T_{pi}$$

Comme Cockburn s'intéresse au temps de sélection global du menu, il prend la moyenne pondérée du temps de sélection pour chaque item. La probabilité que l'item i soit sélectionné est P_i . Ainsi, le temps de sélection moyen T_{avg} s'exprime ainsi :

$$T_{avg} = \sum_{i=1}^n p_i (T_{rdi} + T_{pi})$$

Temps de pointage

Le temps de pointage découle de la loi de Fitts [Fitts 54] (définie précédemment) qui dépend de la largeur et de la distance d'une cible. La largeur pour la loi de Fitts correspond en fait à la hauteur W_i d'un item car le mouvement de pointage est vertical dans les menus linéaires. Sa distance est notée D_i .

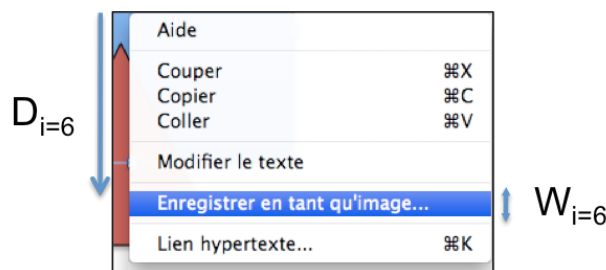


Figure 4 : Largeur et distance d'un item pour la loi de Fitts.

$$T_{pi} = a + b \times \log_2 \left(1 + \frac{D_i}{W_i} \right)$$

Dans le cas général, où tous les items ont la même taille W , alors $D_i = (i-1) \times W$ et :

$$T_{pi} = a + b \times \log_2(i)$$

Temps de recherche/décision

Le temps de recherche/décision T_{rdi} est calculé comme une interpolation du temps de recherche visuelle (T_{rvi}) et du temps de prise de décision de Hyck-Hyman (T_{hhi}).

$$T_{rdi} = (1 - e_i) T_{rvi} + e_i \times T_{hhi}$$

Avec :

- $T_{rvi} = a_{rv} + b_{rv} \times n$ pour un utilisateur inexpérimenté, le temps de recherche visuelle est linéaire en fonction du nombre d'items.
- $T_{hhi} = a_{hh} + b_{hh} \times \log_2 \left(\frac{1}{P_i} \right)$ pour un utilisateur expert, le temps de prise de décision pouvant être prédit par la loi de Hick-Hyman. Il dépend de la probabilité que l'item apparaisse.
- e_i est l'expertise de l'utilisateur. Ainsi, plus l'utilisateur est expert (e_i est proche de 1), plus le temps de recherche visuelle est réduit au temps de prise de décision.

Expertise

L'expertise (e_i) de l'utilisateur pour l'item i dépend du nombre de fois que l'utilisateur a déjà accédé à la commande (t_i) et à la faculté d'apprentissage de l'interface (L) :

$$e_i = L \times \left(1 - \frac{1}{t_i}\right)$$

Cette formule traduit le fait que plus l'utilisateur a sélectionné un item, plus il est considéré comme expert.

Menus hiérarchiques

Lorsque le curseur est sur un item hiérarchique, le sous-menu correspondant s'ouvre (généralement sur la droite). L'utilisateur doit rester sur cet item⁵ pour atteindre ce sous-menu, sinon il risque de le refermer. Le temps de pointage d'un sous-menu peut donc être modélisé par la loi de steering [Accot 97] (steering law). Ainsi, lorsque l'utilisateur a besoin de naviguer à travers les sous-menus pour sélectionner un item i , la formule est la suivante :

$$ML_i = \sum_{j=1}^{i-1} (T_j + sc_j) + T_i$$

où T_j représente le temps moyen pour sélectionner un item au niveau j et sc_j , le temps pour accéder au sous-menu $j+1$. sc_j s'exprime ainsi :

$$sc_j = a_{sc} + b_{sc} \left(\frac{A_j}{W} + 1 \right)$$

où W est la largeur dans la loi de Steering qui correspond en fait à la hauteur de l'item et, A la longueur dans la loi de Steering qui correspond à la largeur de l'item.

En résumé, cette modélisation permet de prédire les performances du mode novice des menus linéaires en fonction de leur organisation et de l'expertise de l'utilisateur. Elle a été vérifiée expérimentalement et a permis de concevoir une nouvelle technique de menus appelée Morphing menus [Cockburn 07] qui élargit les items fréquemment utilisés. Nous retenons les trois sous-tâches explicitées dans cette modélisation (recherche visuelle, tâche de pointage d'un item, tâche de pointage pour accéder à un sous-menu) et le fait que, comme pour SimCIS, le modèle de Cockburn ne considère pas d'autres critères comme l'apprentissage. De plus, ce modèle ne traite pas du mode expert des menus linéaires, ni des menus circulaires.

1.3 Taxonomie des menus 3D

Une taxonomie des « menus 3D », reposant sur celle de Jackoby [Jackoby 92], est présentée dans [Dachselt 07]. Contrairement à notre définition (chapitre 2), le terme « menu » désigne ici tous systèmes de commandes graphiques (menus et palettes). L'étude est centrée sur les menus dans les environnements de réalité augmentée (RA) et réalité virtuelle (RV). Dachselt s'intéresse à ces systèmes de commandes car le contrôle d'application est un problème difficile dans les domaines de la RV et RA. En effet, la 3^{ème} dimension et l'absence ou les limitations des dispositifs [Dachselt 07] peuvent rendre la sélection de commandes complexe. Des solutions sont illustrées Figure 5. Notons que bien que les environnements considérés soient en 3 dimensions, les menus ne sont pas forcément représentés en 3D.

⁵ Aujourd'hui dans la plupart des menus linéaires, un délai est utilisé afin que l'utilisateur puisse déborder de l'item et faire des petites erreurs de « Steering ». Cockburn considère seulement les menus linéaires qui n'utilisent pas ce délai. Nous revenons sur l'importance de ce délai dans le chapitre suivant.

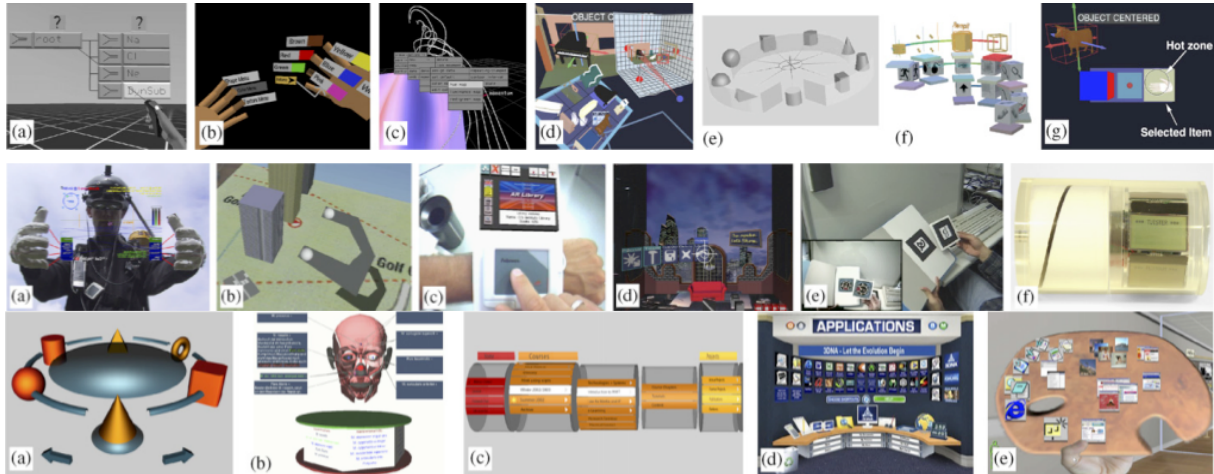


Figure 5 : 18 « menus 3D » pour les environnements de RA et VR. Illustrations extraites de [Daschselt 07].

La taxonomie de la Figure 6 s'appuie sur une liste de critères (colonne de gauche) qui permet de classer les propriétés (colonne de droite) des « menus 3D ». Nous décrivons ci-après les critères utilisés.

- Intention d'utilisation (« intention of use »). Ce critère décrit les menus 3D en répondant aux questions : « Qu'est ce que les développeurs veulent que les utilisateurs choisissent dans le menu ? ». Les propriétés associées sont : *le nombre d'items affichés* qui correspond au nombre de commandes que peut contenir le menu et *la nature hiérarchique* qui différencie un menu (« temporary option menu »), d'une palette (« single menu »), d'une combinaison de menus (« menu system ») et de menus hiérarchiques (« menu hierarchy »).
- Apparence et structure (« Appearance and Structure »). Ce critère décrit la représentation graphique du menu 3D ainsi que son contenu. Les propriétés sont : *la forme géométrique* du menu (sphère, cylindre, etc.), *l'arrangement des items* (en liste, liste circulaire, matrice, etc.), le *type de données* (objets 3D, texte, icônes, etc.) et *la taille et l'espace entre les items*.
- Placement (« Placement »). Ce critère décrit le placement du menu dans l'environnement 3D. Les propriétés sont : la *référence* qui détermine le référentiel dans lequel est positionné le « menu 3D » (un objet du monde 3D, le monde 3D, une partie du corps de l'utilisateur ou un dispositif); *l'orientation* du menu par rapport à la caméra du mode 3D et le *repositionnement* qui permet à l'utilisateur de déplacer le menu afin d'éviter des problèmes d'occlusion.
- Activation et disponibilité (« Invocation et availability »). Il décrit la manière dont l'utilisateur doit activer (ouvrir) le système de commandes. Les propriétés sont : la *visibilité* (« Visibility ») qui distingue si le système est affiché temporairement ou non; *l'activation* (« Invocation ») qui distingue si le menu peut être ouvert n'importe où, seulement à partir d'une icône, d'un autre « menu » ou s'il est simplement tout le temps visible ; *l'animation*, « qui pourrait aussi être associé au critère *apparence et structure* », décrit, comme son nom l'indique, les animations du menus (zoom, disparition progressive, rotation, ...) ; *Collapsibility* indique si le système de commandes peut être iconifié ou masqué temporairement.
- Interaction et dispositifs d'entrées/sorties (« Interaction and I/O setting »). Ce critère est décomposé en 5 propriétés : *le dispositif d'interaction* qui décrit le type de dispositif utilisé pour le « menu 3D » ; *le type d'applications et de dispositifs* (RA, RV, bureau de RV, dispositif mobile 3D) ; la *dimension* qui correspond au *degré d'intégration* [Beaudouin-Lafon 00], c'est à dire la corrélation entre le nombre de dimensions nécessaire pour interagir avec le « menu 3D » et le nombre de dimensions du dispositif d'entrée ; le *retour visuel* utilisé pour la sélection d'un item (surbrillance, animation, couleur, taille, forme) ; et la *visibilité du chemin* qui décrit si le menu affiche le chemin traversé dans une hiérarchie de menus.

- Utilisabilité (« usability »). Ce critère doit être mis en relation avec les critères précédents. Les propriétés sont : les *critères d'évaluation* qui sont des critères utilisés lors d'évaluations expérimentales comme la rapidité, le taux d'erreurs, la facilité d'utilisation, la satisfaction, la facilité d'apprentissage et le confort. Enfin, les *comparaisons* seraient des informations sur les comparaisons entre menus 3D déjà existantes.

- Combinaison (« combinability ») décrit si les « menus 3D » peuvent être combinés entre eux. Aucune propriété n'est fournie pour ce critère.

Intention of use	
<i>Number of displayed items</i>	limited or not, range or definite value
<i>Hierarchical nature</i>	temporary option menu, single menu, menu system, menu hierarchy
Appearance and Structure	
<i>Geometric structure</i>	none, list, disc, sphere, cylinder, cube...
<i>Structural layout</i>	acyclic list, cyclic list (ring), matrix, free arrangement, geometric structure
<i>Type of displayed data</i>	3D-objects, text entries, images, images & text, 3D-objects & text
<i>Size & spacing of items</i>	
Placement	
<i>Reference</i>	world, object, head, body, device
<i>Orientation</i>	
<i>Repositioning</i>	
Invocation and availability	
<i>Visibility</i>	whole time, temporarily, user-dependent
<i>Invocation</i>	icon/miniature, context dependent, free, none
<i>Animation</i>	various ways
<i>Collapsibility</i>	
Interaction and I/O setting	
<i>Interaction device dependence</i>	mouse, gloves, pen & tablet, 6-DOF devices, computer vision, multiple etc.
<i>Application type & setting</i>	VR, AR, Desktop VR, 3D-Mobile
<i>Dimensionality</i>	mapping of interaction device and task
<i>Feedback/highlighting</i>	various ways
<i>Visualization of selection path</i>	
Usability	
<i>Evaluation Criteria</i>	selection speed, error rate, efficiency, user comfort, ease of use and learning
<i>Comparison</i>	of different layouts, selection methods, menu solutions
Combinability	

Figure 6 : Classification des solutions pour les « menus 3D » en fonction de critères (en gras) et de propriétés (en italique). Illustration extraite de [Dachselt 07].

En résumé, cette taxonomie permet d'organiser les systèmes de commandes pour la RA et la RV. Elle est davantage un espace de description qu'un espace de critères car à l'exception de l'utilisabilité, les critères proposés sont des caractéristiques qui permettent de décrire et de classer un menu d'un point de vue système. Aucun résultat n'est donné pour indiquer l'impact de ces caractéristiques sur l'utilisabilité. De plus, les propriétés associées à la recherche visuelle sont absentes. Nous retenons la dimension *nombre de commandes* et la propriété *visibilité du chemin*.

1.4 Synthèse

Vis à vis de ces travaux, notre approche est intégratrice. Nous résumons ici les différents éléments qui sont réutilisés dans notre espace de conception, noté MenUA. Présenté dans la section suivante.

Les espaces de critères (section 1.1) fournissent une grande variété d'éléments pour mesurer la qualité des applications interactives. Nous réutilisons la structure hiérarchique en facteurs et critères de McCall [McCall 77] dans notre espace MenUA afin de décrire précisément les avantages et les inconvénients des techniques de menus. Les facteurs de Nielsen (utilisabilité et utilité) [Nielsen 93], ainsi que les critères d'utilisabilité de Shneiderman [Shneiderman 04] servent de cadre fédérateur pour notre étude. Ces espaces disposent d'une large couverture de critères mais proposent rarement des principes pour les mettre en œuvre. Notre espace MenUA vise à guider la conception des menus en identifiant des critères et les différents *aspects* de la technique qui peuvent être améliorés.

Les modèles de performance (section 1.2) considèrent uniquement les critères de rapidité et de précision, mais permettent de prédire les performances de certaines techniques. Ils mettent en évidence les différents éléments de conception sur lesquels le concepteur peut agir pour améliorer ou créer de nouvelles techniques d'interaction. Dans notre espace MenUA, nous nous appuyons sur les travaux de Appert [Appert 07] afin d'analyser et de nuancer les performances d'une technique de menus en fonction de son contexte d'utilisation. Nous réutilisons également les propriétés morphologiques (organisation temporelle, persistance, fusion, développement et factorisation) du modèle de [Appert 07] qui sont applicables aux techniques de menus. Enfin, nous réutilisons les sous-tâches (recherche visuelle, tâches de pointage) du modèle de Cockburn [Cockburn 07] pour organiser nos solutions de conception vis-à-vis des critères de rapidité et de précision. Ces deux modèles de performances [Appert 07, Cockburn 07] s'appuient sur des lois empiriques (la loi de Fitts, de Hick-Hyman et de Steering) qui permettent structurer nos solutions conceptuelles. Notre espace MenUA réutilise les résultats de ces modèles de performance mais vise une couverture de critères plus étendue en considérant l'apprentissage par exemple, sans aller néanmoins jusqu'à l'évaluation quantitative prédictive.

Enfin, par rapport à la taxonomie de Dachsel [Dachsel 07] qui organise les techniques existantes selon des caractéristiques systèmes, notre espace MenUA les organise vis-à-vis de critères liés à l'utilisateur. Néanmoins, notre espace MenUA réutilise la dimension *nombre de commandes* essentielle pour l'étude des menus circulaires et la propriété de *visibilité du chemin* utile pour la navigation.

Ces différents espaces de conception ont des couvertures bien différentes. Du plus général avec les systèmes interactifs [McCall 77, Nielsen 93, Shneiderman 04] au plus particulier avec le mode novice des menus linéaires [Cockburn 07]. Notre espace MenUA ne couvre que les menus tels qu'ils ont été définis dans le chapitre 1. De plus, nous nous limitons dans cette étude aux techniques de menus qui ont une représentation 2D et qui peuvent être utilisées avec un dispositif de pointage (souris, stylet, écran tactile, etc.) ou un clavier. Pour cette raison, nous ne considérons pas les menus utilisant des mouvements dans l'espace [Oakley 07], les menus vocaux [Yin 06] ou les menus audios [Zhao 07].

2 MenUA : structure globale

Nous présentons la structure globale de MenUA en décrivant les différentes dimensions de notre espace de conception qui seront détaillées dans les deux chapitres suivants.

2.1 Description

Comme dans McCall [McCall 07] et Abowd [Abowd 92], nous adoptons une structuration en deux niveaux *facteurs* et *critères*. Les deux facteurs retenus, *l'utilisabilité* et *l'applicabilité*, sont relatifs à *l'acceptation pratique* d'une technique de menus par les utilisateurs [Nielsen 90] :

- **l'utilisabilité** relève de l'usage : elle traite de l'adéquation d'un système de menus par rapport aux capacités cognitives, motrices et sensorielles de l'utilisateur.
- **l'applicabilité**, à connotation plus fonctionnelle, cible une adéquation d'un système de menus par rapport aux besoins applicatifs de l'utilisateur.

Ces deux facteurs ont inspiré le nom de notre espace : **MenUA**. Le « **U** » fait référence à l'« Utilisabilité » et le « **A** » à l'« Applicabilité ».

Nous présentons maintenant chacun de ces deux facteurs, en termes de critères.

Facteur	Définition	Critères
Utilisabilité	Adéquation du système de menus par rapport aux capacités cognitives, motrices et sensorielles de l'utilisateur.	Rapidité et précision
		Apprentissage et mémorisation
		Satisfaction
Applicabilité	Adéquation du système de menus par rapport aux besoins applicatifs de l'utilisateur.	Adéquation à l'application
		Adéquation à la plateforme
		Adéquation à la tâche

Tableau 1 : Décomposition des deux facteurs Utilisabilité et Applicabilité en critères.

2.2 Facteur d'utilisabilité

Critère d'utilisabilité	Définition	Aspects
Rapidité et précision	Efficacité du menu pour permettre à l'utilisateur de sélectionner des commandes.	Activation du menu
		Recherche visuelle et prise de décision
		Sélection d'un item
		Activation d'un item
		Mode expert
Apprentissage et mémorisation	Capacité du menu à permettre à l'utilisateur de pouvoir l'utiliser de manière optimale rapidement et durablement.	Technique
		Organisation
		Mode expert
Satisfaction	Capacité du menu à fournir un sentiment agréable qui résulte de l'accomplissement de ce que l'on souhaite.	Viscéral
		Comportemental
		Réflexif

Tableau 2 : Les 3 critères associés à l'utilisabilité.

L'*utilisabilité* est un concept complexe pour lequel plusieurs critères ont été proposés [Welie 99a, Welie 99b]. Ces critères représentent le second niveau de structuration dans notre espace MenuA. Dans ce manuscrit, nous ré-utilisons les cinq critères d'utilisabilité de Shneiderman [Shneiderman 04] présentés dans la section 1.1 : rapidité, précision, apprentissage, mémorisation et satisfaction.

Nous traitons ensemble les critères rapidité et précision car le nombre d'erreurs est lié à la vitesse. Similairement, apprentissage et mémorisation sont liés car ils dépendent tous les deux de la fréquence d'utilisation. Finalement, le critère satisfaction est particulier parce que c'est une mesure subjective du confort, du plaisir et de l'acceptation à l'utilisation. De plus, il est rattaché aux autres critères. Par exemple, un utilisateur peut apprécier un menu parce qu'il est rapide. Nous retenons donc les trois critères : *Rapidité et Précision*, *Apprentissage et Mémorisation* et enfin *Satisfaction*.

Une originalité de notre espace de conception est d'introduire un niveau de structuration supplémentaire appelé *Aspect*. Un aspect est un trait commun à tous les menus sur lequel nous pouvons agir vis-à-vis d'un critère. Par exemple, le critère *rapidité et précision* va dépendre de 5 aspects comme la "recherche visuelle" ou la "sélection d'un item".

Le niveau de structuration suivant correspond aux *solutions*. Une solution permet d'améliorer un critère. Elle peut être une propriété comme la "géométrie circulaire", ou une stratégie comme le fait de "augmenter la largeur des items". Les solutions sont organisées selon les différents *aspects* étudiés afin d'expliquer précisément leurs rôles et leurs impacts sur les critères. Par exemple, "augmenter la largeur des items" a un impact sur le critère *rapidité et précision* à travers l'aspect "sélection d'un item".

Enfin, le dernier niveau de structuration correspond aux *techniques de menus*. Les techniques de menus sont composées de plusieurs solutions qu'elles illustrent.

Le facteur d'utilisabilité fait l'objet du chapitre suivant.

2.3 Facteur d'applicabilité

Critère d'applicabilité	Définition	Aspects
Adéquation à l'application	Capacité du menu à contenir les commandes de l'application.	Nombre d'items en Largeur
		Profondeur du système de menus
		Nombre total d'items
Adéquation à la plateforme	Capacité du menu à fonctionner avec différents dispositifs d'entrée et sortie.	Dispositifs d'entrée
		Dispositifs de sortie
Adéquation à la tâche	Capacité du menu à s'intégrer efficacement dans la tâche de l'utilisateur.	Contexte visuel
		Transitions

Tableau 3 : Les trois critères associés au facteur d'applicabilité.

Le facteur *applicabilité*, second facteur de qualité de MenUA, est une transposition du facteur d'utilité de Nielsen [Nielsen 90] aux techniques de menus. Il se différencie de *utilisabilité* dans le sens où il est davantage lié à des contraintes du système et du logiciel : par exemple, le "nombre total d'items" que le menu peut contenir réfère à l'applicabilité du menu pour l'application. Ce point est important car une limitation du nombre de commandes est une forte contrainte logicielle qui peut rendre le menu inutilisable pour certaines applications qui ont un grand nombre de commandes. Similairement, des contraintes du système physique comme la taille de l'écran des dispositifs mobiles peut rendre certaines techniques inappropriées.

L'applicabilité est évidemment liée à l'utilisabilité. Par exemple, le nombre de commandes a un impact sur la rapidité. Plus il augmente, plus le temps nécessaire pour sélectionner une commande augmente. Cependant, à partir d'un certain nombre de commandes, certaines techniques d'interaction ne sont plus considérées comme acceptable. Par exemple, les menus circulaires sont plus rapides que les menus linéaires lorsque le nombre de commandes est inférieur à 8-12 items. Au-delà de ce nombre, les secteurs angulaires sont trop fins et la valeur ajoutée de la technique disparaît : elle perd son avantage concurrentiel. Les critères d'applicabilité offre une vue macroscopique sur les techniques soulignant leurs limites d'utilisation, c'est-à-dire la configuration à partir de laquelle les performances d'utilisabilité chutent et la technique ne devient plus intéressante. A l'opposé, les critères d'utilisabilité mesurent les performances d'une technique pour une utilisation adaptée. Ainsi, ces deux facteurs sont complémentaires car ils étudient les menus à deux niveaux de granularité différents.

Comme pour le facteur d'utilisabilité, l'applicabilité est affinée en quatre niveaux de structuration : critère, aspect, solution et exemple de techniques de menus. Les trois critères retenus sont :

- *l'adéquation à l'application*. Ce critère réfère à la capacité du menu à contenir les commandes de l'application ;
- *l'adéquation à la plateforme*. Ce critère réfère à la capacité du menu à fonctionner correctement avec différents types de dispositifs d'entrée-sortie ;
- *l'adéquation à la tâche*. Ce critère réfère à la capacité du menu à s'intégrer efficacement dans la tâche de l'utilisateur.

Le facteur d'applicabilité fait l'objet du chapitre 5.

3 Conclusion

Nous avons introduit un espace de conception pour les techniques de menus appelé MenUA. Il repose sur une liste de critères organisés en deux facteurs : *l'utilisabilité* et *l'applicabilité*. Nous avons présenté une vue globale en cinq niveaux de structuration (tableau 4) : facteur, critère, aspect, solution et technique de menus.

Facteur	Critères	Aspects	Solutions	menus
Utilisabilité	Rapidité et précision	Activation du menu	Chapitre 4	
		Recherche visuelle et prise de décision		
		Sélection d'un item		
		Activation d'un item		
		Mode expert		
	Apprentissage et mémorisation	technique		
		Organisation		
		Mode expert		
	Satisfaction	Viscéral		
		Comportemental		
Réflexif				
Applicabilité	Adéquation à l'application	Nombre d'items en largeur	Chapitre 5	
		Profondeur du système de menus		
		Nombre total d'items		
	Adéquation à la plateforme	Dispositif d'entrée		
		Dispositif de sortie		
	Adéquation à la tâche	Contexte visuel		
		Transitions		

Tableau 4. Synthèse des différents niveaux de structuration de l'espace MenUA. Du plus abstrait à gauche, au plus concret à droite

L'espace MenUA est détaillé dans les deux chapitres suivants. Le chapitre 4 est dédié au facteur d'utilisabilité et le chapitre 5 au facteur d'applicabilité. Une fois l'espace MenUA présenté, nous concluons sur les éléments moins étudiés, afin de mettre en évidence l'aspect génératif de MenUA. Enfin, La seconde partie de ce mémoire sera consacrée aux techniques de menus créées à partir de MenUA.

Utilisabilité des techniques de menus

Chapitre 4



1	RAPIDITE ET PRECISION	52
1.1	ACTIVATION DU MENU	53
1.2	RECHERCHE VISUELLE ET PRISE DE DECISION	53
1.2.1	FILTRAGE DE COMMANDES	54
1.2.2	MISE EN EVIDENCE DES COMMANDES FREQUENTES	54
1.2.3	GROUPEMENTS INTERNES	56
1.2.4	VOCABULAIRE	57
1.2.5	SOUS-MENUS ET MENUS PARENTS	57
1.3	SELECTION D'UN ITEM	58
1.3.1	REDUCTION DE LA DISTANCE	59
1.3.2	AUGMENTATION DE LA LARGEUR	60
1.3.3	POINTAGE SEMANTIQUE ET POINTAGE D'OBJETS	62
1.3.4	UTILISATION DU RYTHME	65
1.4	ACTIVATION D'UN ITEM	66
1.5	MODE EXPERT	66
2	APPRENTISSAGE ET MEMORISATION	68
2.1	MEMOIRE	68
2.2	APPRENTISSAGE DE LA TECHNIQUE	70
2.3	APPRENTISSAGE ET MEMORISATION DE L'ORGANISATION	71
2.3.1	STRUCTURE DU MENU	71
2.3.2	VOCABULAIRE	71
2.3.3	STABILITE DANS LE TEMPS	71
2.3.4	STABILITE ENTRE APPLICATIONS	72
2.3.5	PERSONNALISATION	72
2.3.6	NAVIGATION	72
2.3.7	ORIENTATION	72
2.4	APPRENTISSAGE ET MEMORISATION DU MODE EXPERT	73
2.4.1	RACCOURCIS CLAVIERS	73
2.4.2	INTERACTION GESTUELLE	74
3	SATISFACTION	76
3.1	EXPERIENCE UTILISATEUR	77
3.2	EMOTION	77
3.3	SATISFACTION ET MENUS	78
4	CONCLUSION	80

Ce chapitre est consacré au facteur *Utilisabilité* de notre espace MenUA (tableau 1). Nous rappelons qu'il est décomposé en trois *critères*, faisant chacun l'objet d'une section.

Facteur	Critères	Aspects
Utilisabilité	Rapidité et précision	Activation
		Recherche visuelle et prise de décision
		Sélection d'un item
		Activation d'un item
		Mode expert
	Apprentissage et mémorisation	technique
		Organisation
		Mode expert
	Satisfaction	Viscéral
		Comportemental
		Réflexif

Tableau 1 : Facteur d'utilisabilité.

La section 1 est dédiée au critère *rapidité et précision*. Nous définissons d'abord un automate général des menus présentant les différentes étapes pour exécuter une commande (Figure 1). Ces étapes, correspondant au processus de sélection dans un menu, définissent les différents *aspects* étudiés pour ce critère. Elles sont utiles pour identifier précisément l'impact des solutions vis-à-vis de ce critère.

La section 2 décrit le critère *apprentissage et mémorisation*. Comme pour le critère *rapidité et précision*, nous proposons trois phases nécessaires pour utiliser un menu de manière optimale qui définissent les aspects étudiés pour ce critère.

Enfin, la section 3 traite du critère *satisfaction*, un critère moins étudié pour lequel nous proposons des éléments d'étude.

1 Rapidité et précision

		Aspects	Solutions
Facteur d'utilisabilité	Critère de rapidité et précision	Activation du menu	
		Recherche visuelle et prise de décision	Filtrage de commandes Mise en évidence des commandes fréquentes (ordre spatial et temporel, taille, couleur) Stabilité de l'organisation Groupements internes et hiérarchiques Vocabulaire Prévisualisation Visibilité du chemin
		Sélection d'un item	Réduction de la distance des items Organisation circulaire Augmentation de la largeur des items Facilité d'accès aux sous-menus Bordures impénétrables Indépendance à l'échelle Délai Pointage sémantique Pointage d'objets Rythme
		Activation d'un item	Crossing
		Mode expert	Raccourcis clavier Gestes

Tableau 2 : Tableau récapitulatif des principales solutions existantes pour le critère rapidité et précision.

Le critère *rapidité et précision* est étudié à travers les quatre étapes de l'automate de la Figure 1. Cet automate, général aux menus, illustre le processus mis en jeu pour l'*exécution d'une commande*. La première étape, *l'activation du menu*, consiste à ouvrir le menu à partir d'un objet d'intérêt, d'une barre de menus ou de tout autre interacteur. Une fois le menu ouvert, l'utilisateur *recherche visuellement* l'item correspondant à la commande désirée. Lorsque celui-ci a été identifié, il *sélectionne l'item*, c'est-à-dire qu'il se positionne sur celui-ci. Ceci a normalement pour effet de mettre l'item en évidence (par exemple en surbrillance). Enfin, il *active l'item* afin d'exécuter la commande (en pressant, relâchant ou cliquant selon les cas). Si l'item est hiérarchique (item donnant accès à un sous-menu), l'utilisateur continue de naviguer dans le système de menus en retournant sur l'état « Menu Ouvert ». Une alternative consiste à utiliser le *mode expert* afin de court-circuiter les étapes précédentes et d'exécuter directement la commande finale.

Ces quatre étapes définissent les différents *aspects* utilisés pour étudier les solutions associées au critère *rapidité et précision*. Ces aspects font chacun l'objet d'un paragraphe ci-après.

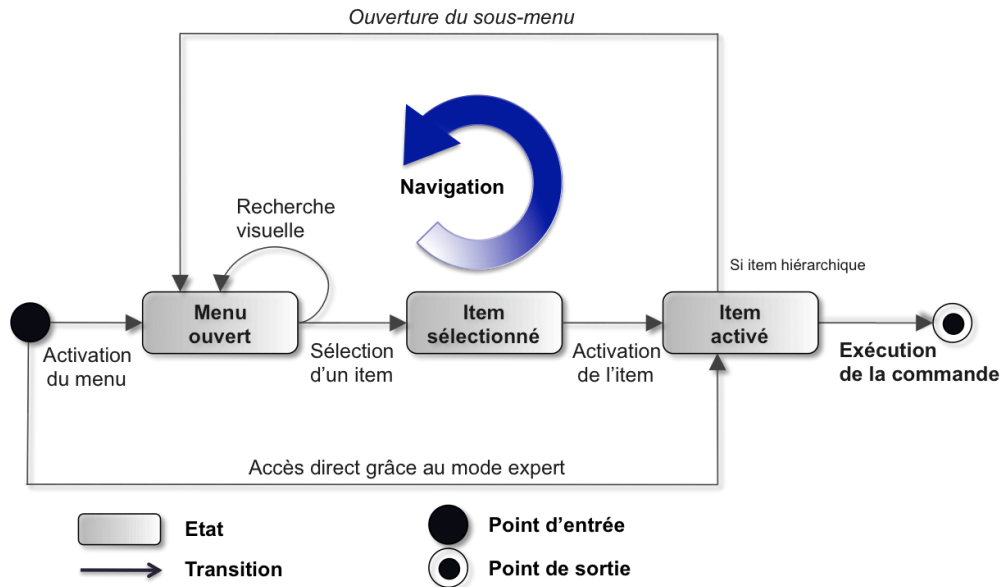


Figure 1 : Processus pour l'exécution d'une commande dans un menu.

1.1 Activation du menu

L'activation (ou invocation [Dachselt 07]) d'un menu consiste à l'afficher et à passer dans un mode où les événements de l'application sont redirigés vers le système de menus considéré (section 1.1 du chapitre 2). Elle se fait habituellement en pressant ou en cliquant avec le bouton droit de la souris mais certains menus n'acceptent que l'un de ces deux types d'activation. Les premiers menus, les « Spring-loaded » menus, et la plupart des menus à base de gestes (comme les Marking menus) sont activés uniquement sur « l'appui » car ils fonctionnent en mode appuyé (« dragging state »). Au contraire, certains menus sous Windows sont activés uniquement sur un clic.

Certains dispositifs, comme les stylets, ne disposent pas d'un bouton droit pour activer les menus. Les solutions à cette limitation, telles que l'utilisation d'un délai sont présentées dans le chapitre suivant, lorsque nous étudions l'adéquation d'une technique de menus à une plateforme d'exécution.

Enfin, certains menus n'apparaissent pas dès le moment où ils sont activés : par exemple, les Marking menus [Kurtenbach 91] sont affichés après un certain délai.

1.2 Recherche visuelle et prise de décision

Avant de sélectionner un item, l'utilisateur doit d'abord effectuer une *recherche visuelle*. Celle-ci consiste à parcourir du regard les différents items du menu pour trouver celui désiré. Dans le chapitre précédent (section 1.2 du chapitre 3) nous avons expliqué que le temps de recherche visuelle pour un item est linéaire en fonction du nombre d'items. Cependant, lorsque l'utilisateur sait à l'avance où se situe cet item, la recherche visuelle se réduit à de la prise de décision (section 1.2 du chapitre 3). Celle-ci est alors prédite par la loi de Hick-Hyman [Hick 52, Hyman 53] qui dépend de la probabilité d'apparition de l'item (nombre d'items et fréquence).

Deux solutions découlent directement de ces observations : *le filtrage de commandes* et *la mise en valeur des commandes fréquentes*. Nous étudions ensuite comment les groupements internes peuvent éviter que l'utilisateur lise la totalité des items, mais seulement un ensemble bien choisi. Enfin, nous traitons le cas des structures hiérarchiques.

1.2.1 Filtrage de commandes

Filtrer les commandes consiste à afficher uniquement un sous-ensemble des commandes disponibles. Par exemple, les menus contextuels ne présentent que les commandes qui peuvent être exécutées sur l'objet sélectionné. Une autre solution courante consiste à "griser" les items qui ne peuvent pas être activés.

Les utilisateurs sont généralement forcés de parcourir la plupart des libellés visibles pour faire leur choix. Une structure hiérarchique permet de décomposer une longue liste de commandes en plusieurs sous-menus. L'utilisateur peut alors effectuer une recherche visuelle uniquement dans quelques sous-menus en "filtrant" les branches de l'arbre de commandes qu'il ne souhaite pas visiter et réduire le nombre d'items à parcourir visuellement.

1.2.2 Mise en évidence des commandes fréquentes

Les observations sur les usages des menus montrent souvent que les deux ou trois items les plus fréquents dans un menu représentent 70 à 90% des sélections [Sears 94]. C'est pourquoi, diverses techniques tentent d'améliorer la recherche visuelle de ces items. Cette stratégie a cependant l'inconvénient de dégrader la recherche visuelle des items moins fréquents et de rendre la représentation visuelle spatialement instable. Une organisation stable, c'est-à-dire une organisation où la position des items n'évolue pas au cours du temps, est plus facile à apprendre et à mémoriser (section 2.3.3). Nous distinguons donc les stratégies qui modifient l'organisation du menu, de celles qui préservent la stabilité spatiale.

Mise en évidence sans préservation de la stabilité

La première stratégie consiste à déplacer les items fréquents en haut du menu afin qu'ils soient visualisés en premier. Cette stratégie repose sur l'hypothèse que l'utilisateur inexpérimenté lit les items des menus linéaires séquentiellement de haut en bas. Aussi, les *Frequency-ordered menus* [Lee 04] déplacent les items les plus fréquents en haut du menu. Plus l'item est utilisé, plus il est haut dans le menu. Les *Frequency-ordered menus* sont illustrés Figure 2.

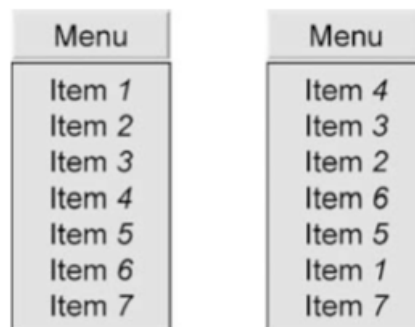


Figure 2 : Les *Frequency-ordered menus* [Lee 04]. Les items 4 et 3 sont les plus utilisés et sont donc déplacés en haut du menu. Illustration extraite de [Lee 04].

La seconde stratégie consiste à masquer temporairement les items moins fréquents. Par exemple, les *Folded menus* [Lee 04], illustrés Figure 3, sont initialement repliés pour laisser apparaître uniquement les commandes les plus fréquentes. Le menu se déplie après un délai ou un clic explicite de l'utilisateur sur le « bouton de dépliage ».

Les *Folded menus* ont été introduits puis retirés des applications Microsoft Office. Une explication vraisemblable est qu'il est difficile de prédire la position des items puisque la stabilité de l'organisation n'est pas vérifiée. De plus, les utilisateurs inexpérimentés sont parfois gênés par le dépliement automatique au moment où ils s'apprêtent à sélectionner un item. Par exemple l'item n°5 en 3^{ème} position sur la Figure 3 change brutalement de position du fait de l'expiration du délai.

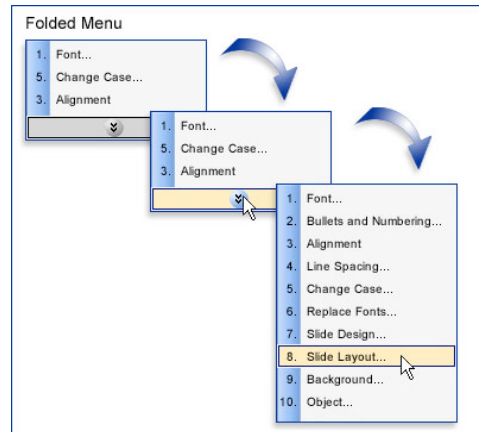


Figure 3 : Folded menus. Initialement replié pour laisser apparaître uniquement les items les plus fréquents, le menu est déplié après un délai ou un clic sur le bouton « dépliage ». (Illustration extraite de [humanf. 04])

Mise en évidence avec préservation de la stabilité

Contrairement aux solutions précédentes, ces techniques ont pour avantage de préserver la stabilité du menu.

La première stratégie consiste à utiliser des attributs visuels qui attirent le regard sur les items fréquents, en augmentant la taille des items (*Morphing menus* [Cockburn 07], Figure 4-a), en changeant la couleur de fond (*Bubbling menus* [Tsandilas 07], Figure 4-b), en ajoutant des icônes colorées (Figure 4-c) ou encore en modifiant la transparence des items. Les menus translucides [Bowes 03] réduisent la visibilité des items les moins fréquents en augmentant leur transparence. Cependant, la transparence peut entraîner des problèmes de lisibilité comme le montre la Figure 4-d.

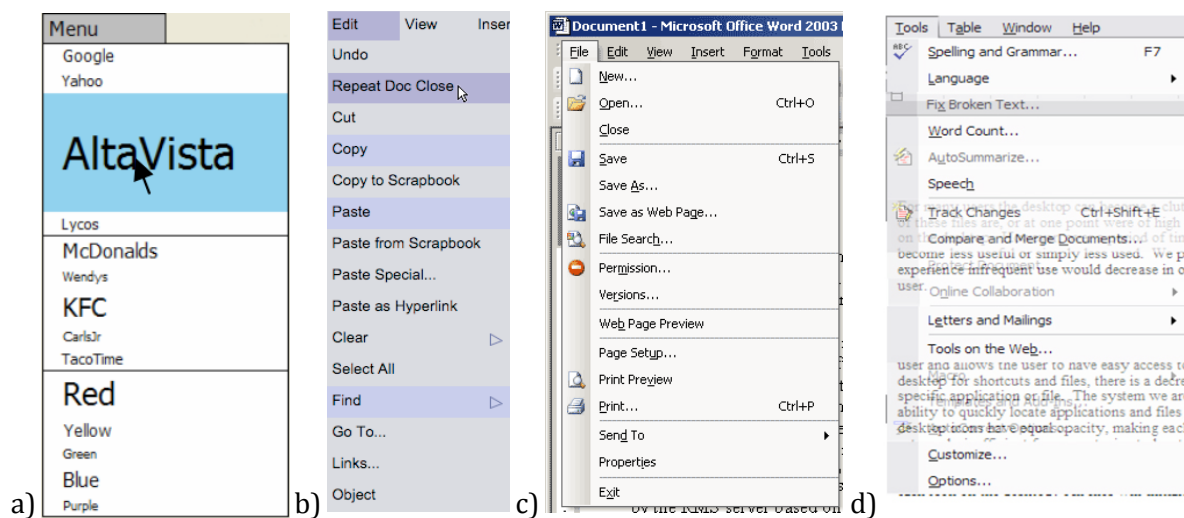


Figure 4. Mise en valeur des items fréquents en exploitant des attributs visuels. A) les *Morphing menus* [Cockburn 07] qui augmentent la hauteur des items, b) les *Bubbling menus* [Tsandilas 07] qui changent la couleur du fond, c) menus linéaires avec des icônes et d) les menus translucides [Bowes 03] qui augmentent la transparence des items les moins fréquents.

La seconde stratégie consiste à ne pas afficher toutes les commandes en même temps [Lee 04]. Les items fréquents sont toujours affichés tandis que les autres n'apparaissent qu'après un délai. Les *Temporal menus* [Lee 04], basés sur ce principe de « priorité temporelle », utilisent un délai de 170ms (Figure 5). A la différence des *Folded menus*, ils ne sont jamais repliés : les zones associées aux items peu fréquents restent vides jusqu'à l'expiration du délai.

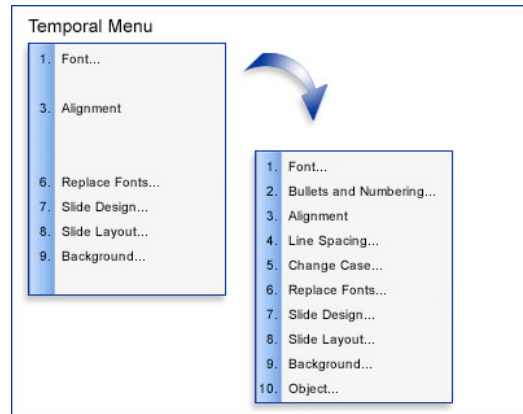


Figure 5 : Les Temporal menus [Lee 04] affichent d'abord les items fréquents sans modifier l'organisation des commandes. Illustration extraite de [humanf. 04].

Enfin, la dernière stratégie consiste à dupliquer les commandes les plus fréquentes comme dans les *Split menus* [Sears 94] où quelques items (2-4) apparaissent à la fois en haut du menu et à leur place habituelle parmi les autres items (Figure 6). Ceci permet, comme pour les Frequency-ordered menus, de remonter les commandes fréquentes tout en conservant une organisation stable. Les Split menus sont utilisés dans les produits Microsoft Office depuis 2000 pour sélectionner les polices de caractères.



Figure 6 : Les Split menus [Sears 94]. Les items les plus fréquents sont dupliqués et positionnés en haut du menu. Illustration extraite de [Sears 94]

1.2.3 Groupements internes

Plusieurs groupements internes, habituellement séparés par une ligne horizontale, peuvent figurer à un même niveau de la hiérarchie d'un système de menu (Figure 7). Ils constituent une alternative aux groupes hiérarchiques pour organiser les commandes sémantiquement. Les groupements internes favorisent la recherche visuelle dans la mesure où la lecture d'un ou deux items d'un groupe peut suffire à l'utilisateur pour estimer si la commande qu'il cherche a ou non des chances de s'y trouver. Ils permettent donc de diminuer le nombre de libellés à lire pour trouver l'item désiré. De plus, le groupement de commandes par proximité sémantique facilite la navigation en guidant l'utilisateur lorsqu'il recherche des commandes proches de commandes dont il connaît déjà l'emplacement (par exemple, s'il connaît déjà "Enregistrer" et recherche "enregistrer sous", il saura dans quelle zone aller).

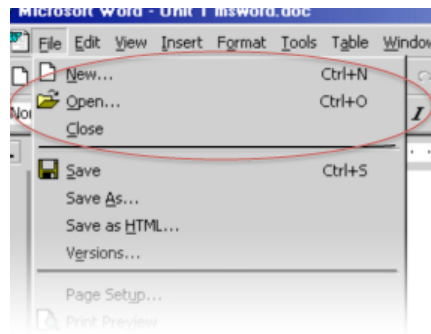


Figure 7 : Groupements internes dans les menus linéaires séparés par une ligne.

Apparemment, peu de travaux semblent traiter des groupements internes dans les menus circulaires. Ceci peut être dû au fait que ce type de menu n'accepte généralement que peu de commandes à un niveau donné de la hiérarchie. Nous reviendrons sur ce point dans le chapitre 7 en proposant une façon originale de mettre en valeur les groupements internes dans les menus circulaires.

1.2.4 Vocabulaire

Le vocabulaire utilisé joue un rôle essentiel pour la recherche visuelle, et plus largement, pour les tâches de navigation. Souvent, c'est davantage une fonctionnalité qu'une commande précisément identifiée que l'utilisateur recherche (chapitre 2, section 3). Des libellés compréhensibles avec une terminologie cohérente facilitent l'identification de l'item désiré [Norman 91]. De plus, dans les systèmes de menus, le libellé des items hiérarchiques doit refléter les commandes du sous-menu et ne pas évoquer des commandes se trouvant dans d'autres sous-menus [Norman 91].

Un compromis est souvent nécessaire sur la taille des libellés. Des libellés longs transmettent davantage d'information tandis que des libellés plus petits sont plus rapides à lire et occupent moins de place à l'écran.

1.2.5 Sous-menus et menus parents

Lorsque l'utilisateur navigue dans une structure hiérarchique, il est souvent amené à rechercher un item dans les sous-menus ou dans les menus parents lorsqu'il est descendu dans une mauvaise branche.

La *prévisualisation* favorise la recherche visuelle dans les sous-menus. Cette fonctionnalité consiste à ouvrir automatiquement un sous-menu lorsque le curseur est sur un item hiérarchique¹. Elle est présente dans la plupart des menus linéaires : après un bref délai, les sous-menus sont affichés automatiquement sans nécessiter d'action explicite de l'utilisateur. La prévisualisation est très utile pour la recherche visuelle car elle permet d'explorer un ensemble de sous-menus juste en déplaçant la souris dans le menu courant. Elle offre à l'utilisateur la possibilité de les visualiser successivement et continuellement les sous-menus sans se déplacer dans la hiérarchie ce qui nécessiterait un bien plus grand nombre d'actions qu'un simple déplacement de la souris. Malgré son intérêt, cette fonctionnalité n'a pas reçu beaucoup d'attention dans la littérature sur les menus, en particulier en ce qui concerne les menus circulaires [Forström 06].

Inversement, la *visibilité du chemin* [Dachselt 07] facilite la recherche d'un item dans les menus parents. Elle consiste à laisser tous les menus parents² visibles à l'écran. Cette solution

¹ Un item hiérarchique est un item qui donne accès à un sous-menu (section 1 du chapitre 2).

² Les menus parents sont tous les menus qui ont permis d'accéder au menu courant (section 1 du chapitre 2).

est utilisée par la plupart des menus linéaires, mais plus rarement par les menus circulaires. Nous détaillons ces deux solutions qui facilitent la navigation dans les systèmes de menus dans le chapitre 6.



Nous retenons que la recherche visuelle peut être améliorée en filtrant les commandes, en les mettant en évidence, en les groupant et en utilisant un vocabulaire cohérent. En outre, la prévisualisation et la visibilité du chemin facilitent la navigation dans les systèmes de menus.

1.3 Sélection d'un item

Après avoir identifié un item, l'utilisateur doit le sélectionner. Nous distinguons la *sélection de l'activation d'un item*.

La sélection d'un item consiste à ce que l'item courant soit le bon (section 1 du chapitre 2). Avec un dispositif de pointage, la sélection correspond à une tâche de pointage. L'utilisateur doit déplacer le curseur au dessus de l'item. Ce déplacement peut être réalisé avec le bouton de la souris relevé ou enfoncé. Avec un clavier, l'utilisateur utilise le pavé directionnel ou les touches mnémoniques (section 1 du chapitre 2). Il en résulte sa mise en évidence (par exemple surbrillance). Contrairement à l'activation d'un item, étudiée dans la section suivante, la sélection d'un item ne déclenche pas l'exécution d'une commande, elle modifie uniquement l'état interne du menu.

Nous distinguons trois types de sélection pour les menus hiérarchiques :

- dans le menu courant ;
- dans le sous-menu ;
- dans un menu parent.

A notre connaissance, aucun travail ne s'est spécifiquement intéressé à la sélection d'un item dans un menu parent. Comme nous l'avons indiqué au chapitre 2, l'utilisateur est souvent amené à emprunter des chemins détournés et donc à remonter dans la hiérarchie. Du fait de la visibilité du chemin dans les menus linéaires, il est alors aisé de remonter dans la hiérarchie. Cependant, ce n'est généralement pas le cas avec les menus circulaires, les menus parents n'étant pas affichés afin de réduire l'espace occupé à l'écran (section 2 du chapitre 5). Ceci oblige l'utilisateur à remonter pas à pas dans la hiérarchie, en fermant le menu courant à chaque niveau pour pouvoir ouvrir le menu parent.

Concernant la sélection d'un item appartenant au menu courant ou à un sous-menu, des approches similaires sont utilisées comme la réduction de la distance ou l'augmentation de la largeur de la cible. Comme expliqué au chapitre 3, le temps de sélection d'un item est prédit par la loi de Fitts [Fitts54] comme étant une fonction linéaire de l'indice de difficulté : $ID = \log_2(D/W + 1)$. Dans le cas de la sélection d'un item dans un sous-menu, il faut rajouter le temps pour accéder au sous-menu. Celui-ci est mieux prédit par la loi de Steering [Accot 97, Ahlström 05, Cockburn 07] car le curseur doit rester dans l'item parent pour ne pas risquer de fermer le sous-menu. C'est une fonction linéaire du rapport L/W . Ainsi, quelle que soit la loi empirique, il est possible de réduire le temps de sélection en réduisant le paramètre distance ou en augmentant le paramètre largeur. Les sections suivantes décrivent diverses solutions permettant d'améliorer le temps de sélection d'un item qui dérivent de ces lois empiriques.

1.3.1 Réduction de la distance

La technique la plus connue pour réduire la distance moyenne des items est certainement les *Pie menus* [Callahan 88]. Contrairement aux menus linéaires pour lesquels le temps de sélection d'un item varie selon sa position dans le menu, les Pie menus permettent de sélectionner n'importe quel item en temps constant grâce à une organisation circulaire autour du point d'activation. Les Pie menus réduisent de plus de 15% le temps de sélection par rapport à un menu linéaire de 8 items [Callahan 88]. Un Pie Menu est illustré Figure 8



Figure 8 : Pie menu [Callahan 88]. Les items sont organisés circulairement autour du point d'activation. Illustration extraite de [second Life].

Les menus linéaires contextuels apparaissent généralement en bas à droite du point d'activation si la place disponible est suffisante. Cette organisation n'optimise pas la distance à parcourir. Une amélioration consiste alors à placer le menu à mi-hauteur du point d'activation ce qui permet de réduire la distance moyenne des items [Walker 90] (Figure 9). La distance maximale est alors de $D_{\max} = (n-1)*H/2$ au lieu de $D_{\max} = (n-1)*H$ (pour un menu de n items de hauteur H). Cependant, une telle disposition a l'inconvénient de masquer davantage les petites cibles, comme le montre la Figure 9. Ce placement à mi-hauteur a également été utilisé pour positionner les sous-menus dans les menus linéaires [Tanvir 08] (Figure 15).

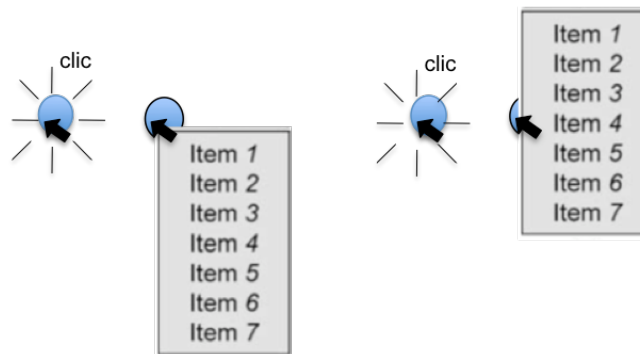


Figure 9 : Réduction en moyenne de la distance des items en centrant les menus linéaires à mi-hauteur.

Une autre solution consiste à réduire la distance des items fréquents. Par exemple, les *Frequency-ordered menus* [Lee 04], les *Split menus* [Sears 04] et les *Folded menus* que nous avons présentés dans la sous-section précédente améliorent également la sélection des items fréquents (en les plaçant en haut du menu). Lee [Lee 04] a comparé les performances de ces menus pour la recherche visuelle et la sélection. Il conclut que les Split menus sont les plus efficaces lorsque les 3 commandes les plus fréquentes sont sélectionnées entre 30.4% et 90.0% du temps. Au dessus de 90.0%, les Folded menus sont meilleurs à condition que la distribution de fréquence soit fixe. En dessous de 30.4%, il n'y a pas d'avantage à utiliser d'autres menus que les menus traditionnels.

Une autre approche consiste à réduire la distance pour accéder aux sous-menus (Figure 10). Kobayashi [Kobayashi 03] rapproche les sous-menus dans les menus linéaires en utilisant la direction du mouvement du curseur. Un mouvement horizontal sur la droite/gauche ouvre/ferme le sous-menu courant à l'endroit où se trouve le curseur. Ainsi, le curseur ne subit

plus la loi de steering car il n'est plus forcé de traverser l'item parent pour accéder au sous-menu. Les mouvements verticaux n'ont pas d'effet autre que celui de mettre en surbrillance l'item qui se trouve en dessous du curseur. Ce menu (illustré Figure 10) a l'inconvénient d'altérer la navigation car les sous-menus masquent partiellement les menus parents ce qui pénalise les utilisateurs qui réalisent un chemin fortement détourné (chapitre 2, section 4).

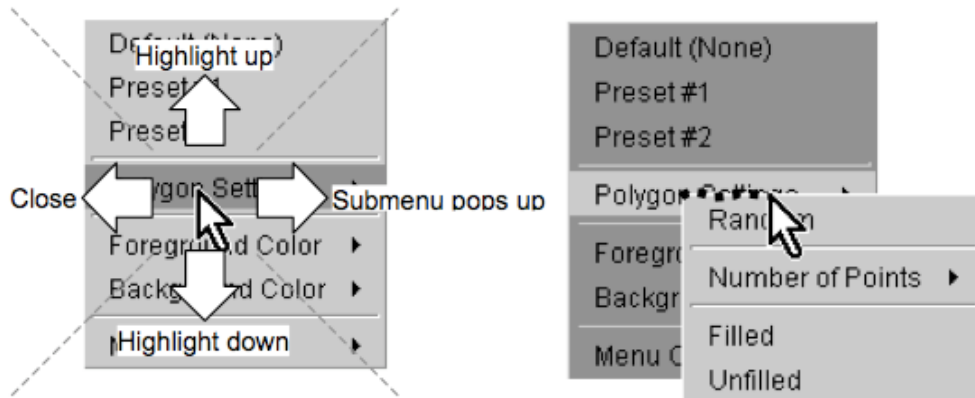


Figure 10 : L'utilisateur réalise un mouvement sur la droite pour ouvrir le sous-menu. Celui-ci s'affiche sous le curseur. Illustration extraite de [Koybayashi 03].

Nous venons de voir différentes stratégies diminuant la distance de la cible pour améliorer le temps de pointage. Nous examinons maintenant les stratégies qui augmentent la largeur de la cible pour la loi de Fitts ou la largeur du « tunnel » pour la loi de steering.

1.3.2 Augmentation de la largeur

La *largeur* de la cible pour la loi de Fitts ou la *largeur du tunnel* pour la loi de steering correspond à la hauteur des items dans le cas des menus linéaires.

Walker [Walker 90] propose un menu linéaire dont la hauteur des items est proportionnelle à leur distance afin de conserver un indice de difficulté constant quel que soit l'item considéré (Figure 11). Une approche moins radicale consiste à augmenter uniquement la hauteur des items fréquents comme dans les *Morphing menus* [Cockburn 07], illustrés Figure 4a.

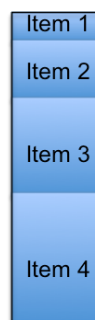


Figure 11 : La hauteur des items augmente proportionnellement à la distance afin de conserver un indice de difficulté constant.

Walker propose aussi des menus linéaires comportant des bords impénétrables [Walker 90] ne permettant pas au curseur de la souris de dépasser le dernier item du menu (ce qui revient à avoir un item de hauteur infinie). Ainsi, l'utilisateur peut faire un mouvement aussi rapide et aussi long qu'il le désire car il ne peut pas dépasser la cible [Appert 08]. Il est à noter que la barre de menus sur les Macintosh repose sur la même idée : elle est affichée tout en haut de l'écran, qui joue le rôle d'un bord impénétrable.

Les *Marking menus* [Kurtenbach 91] laissent l'utilisateur faire des marques indépendantes à l'échelle (« scale indépendance »). L'interaction ne dépend plus de la longueur des marques, comme si les secteurs des Pie menus avaient une taille infinie (Figure 12). La seule

contrainte correspond à l'amplitude angulaire des secteurs, laquelle dépend du nombre d'items. Nous revierons sur ce point dans le chapitre suivant.

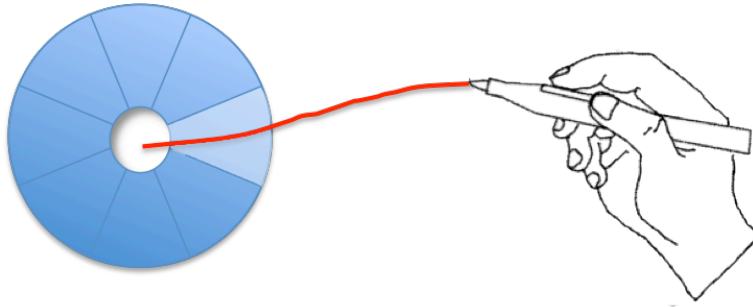


Figure 12 : Indépendance à l'échelle dans les Marking menus [Kurtenbach 91]. La taille des marques n'a pas d'importance.

Pour les menus hiérarchiques, des approches consistent à augmenter la *largeur du tunnel* permettant d'accéder aux sous-menus. Il existe deux variantes, soit temporelle, soit spatiale.

La variante temporelle utilise un délai entre le moment où le curseur rentre (ou sort) d'un item hiérarchique et le moment où le sous-menu s'ouvre (respectivement se ferme). « Ce délai permet de faire de petites erreurs de conduite (steering errors) » [Cockburn 06] lorsque l'on accède au sous-menu en traçant une diagonale (Figure 13). Cockburn [Cockburn 06] affirme cependant que ce délai est souvent trop court pour des utilisateurs inexpérimentés et trop long pour des utilisateurs expérimentés.

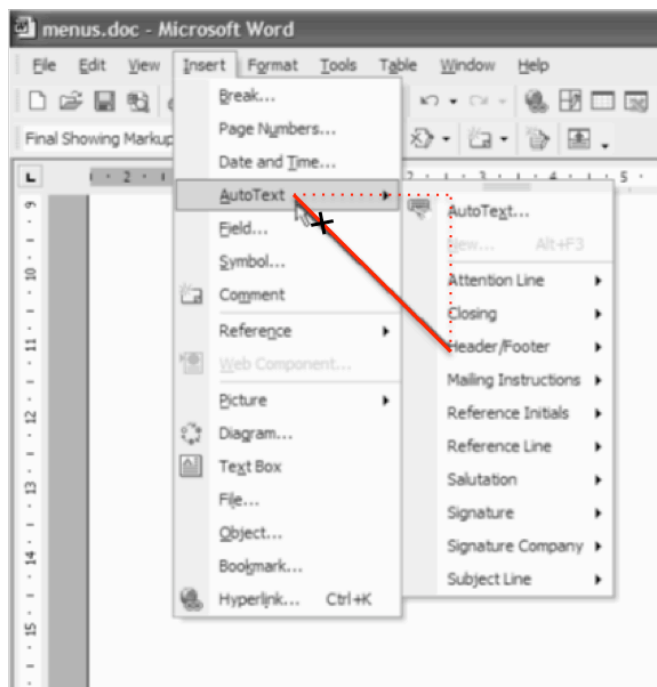


Figure 13 : Le chemin le plus court pour accéder à un item du sous-menu est la diagonale. Sans l'utilisation, d'un délai, ce chemin provoquerait la fermeture du sous-menu. Illustration extraite de [Cockburn 96].

La variante spatiale permet d'augmenter la largeur du tunnel sans utiliser de délai. Les *MenUs* [Cockburn 06], illustrés Figure 14, laissent le sous-menu ouvert tant que le curseur ne rentre pas dans un nouvel item hiérarchique. La zone associée à l'ouverture du sous-menu dépend donc de la densité des items hiérarchiques dans le menu. Dans le pire scénario (où tous les items sont hiérarchiques), la zone associée à l'ouverture du sous-menu est similaire à celle des menus en cascades (c'est-à-dire réduite à celle d'un item hiérarchique). Cependant, cette solution a l'inconvénient de masquer une zone de l'écran plus importante lorsque l'utilisateur ne souhaite pas sélectionner un item dans un sous-menu.

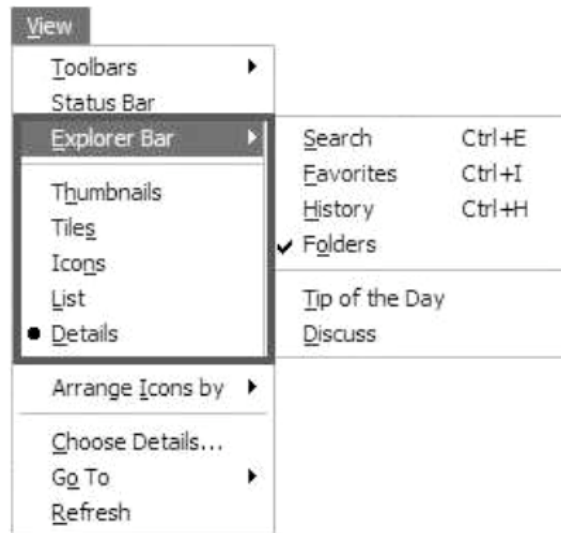


Figure 14 : MenUs [Cockburn 06] laissent le sous-menu affiché tant que le curseur n'est pas dans un nouvel item hiérarchique. La zone d'ouverture du menu est donc élargie. Illustration extraite de [Cockburn 06].

Tanvir [Tanvir 08] propose une autre solution avec les *AAMU menus*. Les AAMU menus (Figure 15) utilisent une zone triangulaire pour faciliter l'accès aux sous-menus. Deux des sommets du triangle correspondent aux extrémités du sous-menu. Le troisième sommet correspond à la position du curseur lorsque le menu a été ouvert. Tant que le curseur reste dans cette zone, le sous-menu reste ouvert. Une différence avec le système précédent (MenUs) est que AAMU est indépendant de la densité des items hiérarchiques car la zone triangulaire se superpose au menu parent. Afin de ne pas altérer la lisibilité et la navigation, cette zone est translucide. De plus, avec AAMU, les sous-menus sont affichés à mi-hauteur de leurs items parents afin de réduire la distance moyenne d'accès aux items (section 1.3.1).

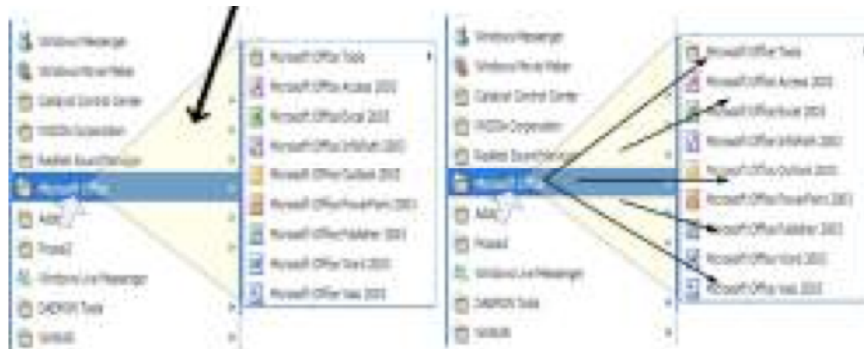


Figure 15 : Les AAMU menus [Tanvir 08]. Introduction d'une zone triangulaire pour faciliter l'accès en diagonale aux sous-menus. De plus les sous-menus sont affichés à mi-hauteur de l'item parent. Illustration extraite de [Tanvir 08].

Nous venons de voir comment améliorer la sélection d'un item en diminuant la distance ou en augmentant la largeur à la fois dans l'espace moteur et dans l'espace d'affichage. Nous étudions maintenant deux solutions, le « pointage sémantique » et le « pointage d'objet » qui différencient ces deux espaces.

1.3.3 Pointage sémantique et pointage d'objets

Le pointage sémantique [Blanch 04b] et le pointage d'objets [Guiard 04] consistent à séparer l'espace moteur de l'espace d'affichage, ceci afin de réduire l'indice de difficulté dans l'espace moteur sans perturber l'espace d'affichage.

Le pointage sémantique

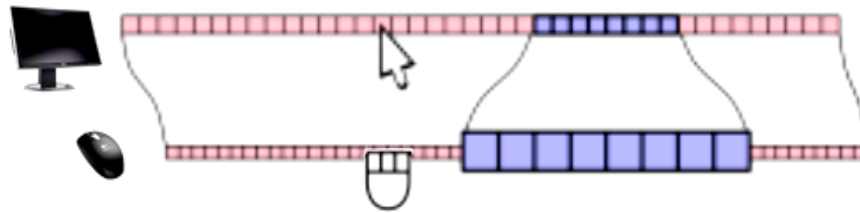


Figure 16 : Pointage sémantique. Le ratio « control/display » comme facteur d'échelle entre l'écran (ligne de pixels en haut) et l'espace de la souris (« images » des pixels sur la table après un changement d'échelle en bas). Illustration extraite de [Blanch 04a].

Le pointage sémantique consiste à adapter dynamiquement le ratio « control-display » (C-D). Le C-D est le rapport entre la distance parcourue par la souris et la distance parcourue par le curseur sur l'écran. Pour un petit C-D, un large mouvement de la souris déplace modérément le curseur. Pour un C-D plus élevé et un mouvement de même amplitude, le curseur parcourt une distance plus grande. Le pointage sémantique consiste à adapter dynamiquement le C-D afin que certaines portions de l'écran soient dilatées ou contractées dans l'espace moteur. En fonction de la sémantique du pixel, le curseur « accélère » dans les espaces vides et « freine » près des cibles potentielles. Par exemple, la Figure 16 montre que les pixels foncés (violet) à l'écran sont dilatés dans l'espace moteur (souris). Le temps de mouvement suit toujours la loi de Fitts dans l'espace moteur, mais plus dans l'espace d'affichage.

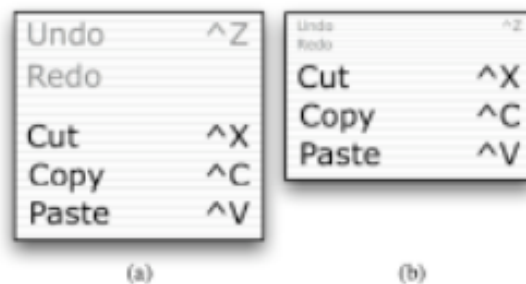


Figure 17 : (a) Version à l'écran, (b) Version dans l'espace moteur qui permet de réduire les entrées inactives. Illustration extraite de [Blanch 04b].

Blanch [Blanch 04b] propose dans le cadre des techniques de menus, d'utiliser le pointage sémantique pour favoriser la sélection des commandes actives (non-grisées), (Figure 17). Ahlström [Ahlström 05] utilise le pointage sémantique pour créer des "champs de force". Les *Force Fields menus*, illustrés Figure 18, appliquent un "champ de force" sur les items hiérarchiques pour attirer le curseur en direction des sous-menus et en faciliter l'accès. De plus, lorsque le curseur est au-dessus d'un item non-hiérarchique, il est ramené au centre de l'item pour éviter les erreurs.

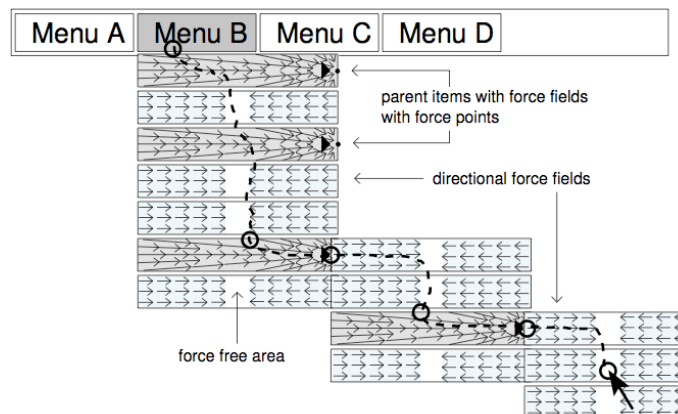


Figure 18 : Les Force Fields menus [Ahlström 05] utilisent un « champs de force » pour favoriser l'accès aux sous-menus et prévenir des erreurs. Illustration extraite de [Ahlström 05].

Le pointage d'objets [Guiard 04] est un cas limite du pointage sémantique qui "supprime" artificiellement l'espace vide entre les objets du bureau, réduisant au maximum la distance entre les objets dans l'espace moteur. Les *Jumping menus* [Ahlström 06] utilisent le pointage d'objets pour accéder à un sous-menu sans devoir réaliser une tâche de pointage. L'utilisateur clique sur l'item parent et le curseur "saute" (warping) directement sur le sous-menu.

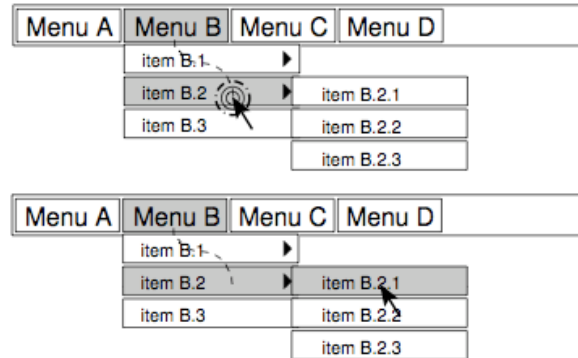


Figure 19 : Les Jumping menus [Ahlström 06] qui permettent de faire sauter le curseur d'un item parent à son sous-menu. Illustration extraite de [Ahlström 06].

Un inconvénient des Force-Field et Jumping menus est que l'utilisateur peut être désorienté de ne pas avoir un contrôle total du mouvement du curseur [Tanvir 08]. En effet, ces menus ne contrôlent pas seulement l'accélération de la souris, mais modifient l'orientation du curseur (dans les Force-Field menus) ou sa position (dans les Jumping menus).

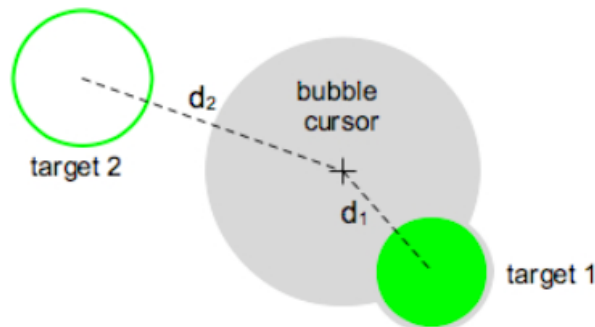


Figure 20 : Le Bubble curseur. La taille du curseur change dynamiquement afin de toujours sélectionner la cible la plus proche ($d_1 < d_2$). Illustration extraite de [Grossman 05].

Enfin, les *Bubbling menus* [Tsandillas 07] favorisent la sélection des items favoris en adaptant dynamiquement la taille du curseur grâce à un Bubble curseur [Grossman 05]. La technique du Bubble curseur consiste à augmenter la taille du curseur afin que celui-ci puisse toujours sélectionner l'élément le plus proche (Figure 20). Dans les Bubbling menus, illustrés Figure 21, les items favoris ont une couleur de fond différente (bleue). Deux modes d'utilisation cohabitent. Si l'utilisateur a le bouton de la souris relevé le menu fonctionne comme les menus traditionnels. Si l'utilisateur maintient le bouton de la souris enfoncé, le Bubble curseur est activé, et saute de favoris en favoris. De plus, le Bubble curseur est désactivé si l'utilisateur revient dans la partie gauche du menu.

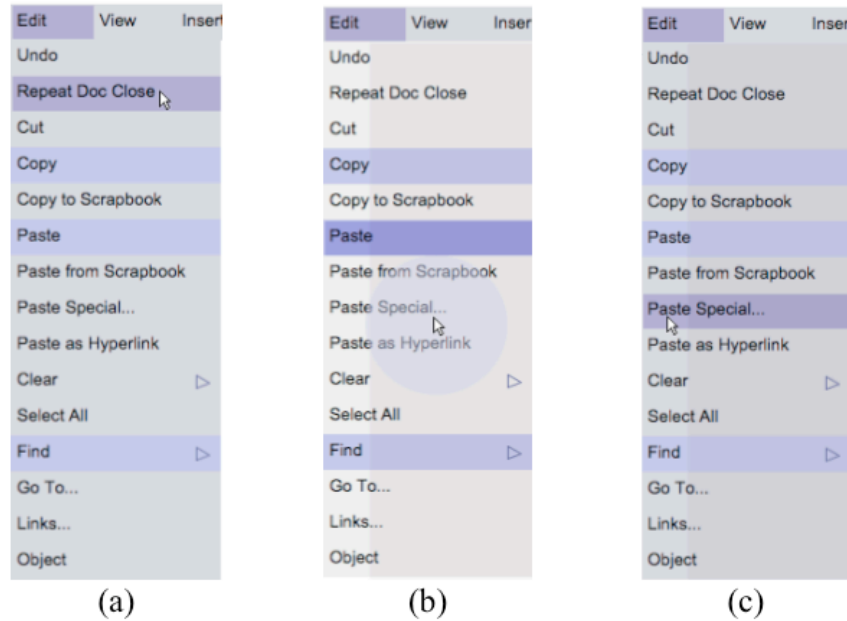


Figure 21 : Les Bubbling menus [Tsandillas 07]. (a) La vue par défaut avec les items favoris en bleu (bouton de la souris relevé). (b) Utilisation du Bubble curseur qui saute de favoris en favoris (bouton de la souris enfoncé). (c) Lorsque le curseur revient dans la partie gauche du menu, le Bubble curseur disparaît. Illustration extraite de [Tsandillas 07].

Une dernière stratégie originale s’émancipe de la dimension spatiale en utilisant le rythme.

1.3.4 Utilisation du rythme

Comme le nom l’indique, les *menus rythmiques* [Maury 99a, Maury 99b] utilisent seulement le rythme pour sélectionner une commande. Une fois le menu ouvert, le système met en surbrillance chaque item successivement à un rythme constant (Figure 22). Lorsqu’un item est en surbrillance (item courant), l’utilisateur peut l’activer en cliquant n’importe où à l’écran. La spécificité de cette technique est que la sélection est déléguée au système. Au lieu que l’utilisateur réalise la tâche de pointage, celle-ci est faite par le système : l’utilisateur n’a plus qu’à confirmer. Maury montre que les menus rythmiques sont plus rapides que les menus classiques lorsqu’ils contiennent moins de 9 items.

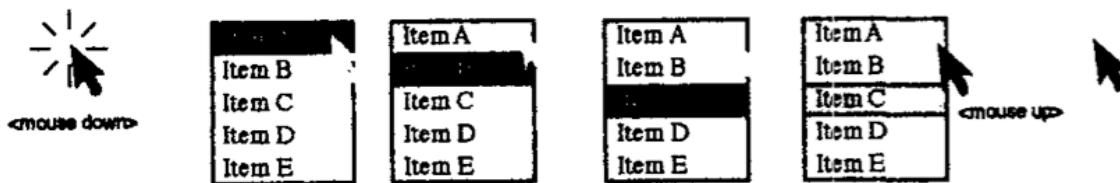



Figure 22 : Les menus rythmiques [Maury 99]. Le système met en surbrillance chaque item à rythme constant. Illustration extraite de [Maury 99a, Maury99b].



Nous retenons la multitude des travaux visant à améliorer la sélection des items dans les menus que ce soit en réduisant la distance à parcourir, en augmentant la taille des cibles, en réduisant les erreurs de steering ou encore en déléguant la tâche au système.

1.4 Activation d'un item

Un item sélectionné peut être activé afin d'exécuter la commande qui lui est associée. Contrairement à la sélection d'item, qui modifie uniquement l'état interne du menu (mise en surbrillance de l'item), l'activation modifie les objets d'intérêt de l'utilisateur. Le menu se referme automatiquement juste avant l'exécution de la commande. Une exception cependant pour les items hiérarchiques qui ouvrent le sous-menu correspondant lorsque qu'ils sont activés.

Dans les menus linéaires, l'activation est provoquée par un clic si le bouton de la souris est relevé et par un relâchement (« release ») si le bouton est déjà enfoncé. Lorsque l'utilisateur utilise le clavier, il active l'item sélectionné en appuyant sur la touche entrée ou la barre d'espace.

Dans les Marking menus, l'interaction est basée uniquement sur l'état « enfoncé » (« dragging state »), c'est-à-dire avec le bouton de la souris enfoncé. La commande est alors exécutée sur le relâchement du bouton de la souris.

Enfin, certaines techniques proposent d'activer un élément en le franchissant [Pook 00, Guimbretiere 00, Apitz 04]. Par exemple, au lieu d'appuyer sur un bouton de l'interface, il suffit que l'utilisateur le traverse de part en part pour exécuter l'opération associée [Apitz 04]. Dans le cas des techniques de menus, l'activation par franchissement a l'avantage de pouvoir exécuter une commande sans relâcher le bouton de la souris. Cette caractéristique est utilisée pour activer une commande et contrôler sa valeur dans le même geste (« merging »), que nous détaillons dans le chapitre suivant (section 3.1).

1.5 Mode expert

Nous avons vu dans le chapitre 2 que le mode expert permettait d'exécuter une commande sans traverser les différents sous-menus de la hiérarchie. Cette fonctionnalité, donnant un *accès direct* aux commandes, est particulièrement efficace car elle supprime toutes les étapes étudiées précédemment (Figure 1) : recherche visuelle, sélection d'un item, et activation de celui-ci. Par exemple, dans les menus linéaires, les raccourcis claviers permettent d'accéder très rapidement aux commandes. Cependant, ils nécessitent que l'utilisateur les apprenne : le mode expert fait appel à la mémoire alors que le mode novice fait appel à la reconnaissance (chapitre 2).

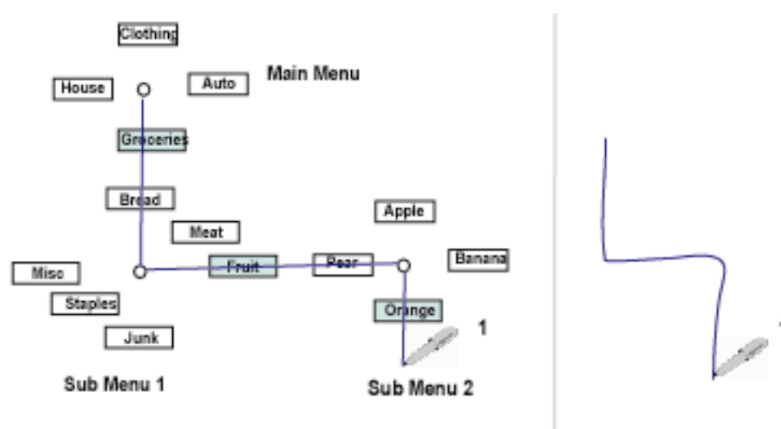


Figure 23 : Dans les Compound Marking menus, la marque en mode expert correspond au chemin à parcourir en mode novice.

Dans les menus circulaires, le concept d'*accès direct* est plus délicat. Par exemple, avec les *Compound Marking menus* [Kurtenbach 93a] (Figure 23), nous pourrions considérer que l'utilisateur traverse également la hiérarchie de commandes en mode expert. En effet, la Figure 23 montre que le chemin parcouru par la souris en mode expert est identique à celui parcouru

en mode novice : les différents niveaux de profondeur peuvent être retrouvés grâce à l'analyse des inflexions. Ainsi, le chemin parcouru en mode expert serait similaire au *chemin droit* parcouru en mode novice et ne constituerait pas à proprement parler un *accès direct* à la commande. Cependant, d'un point de vue utilisateur, cette marque est interprétée comme un "tout" et considérée comme un unique mnème (« chunk ») [Kurtenbach 93a, Buxton 95, Zhao 04]. Un mnème constitue la plus petite unité de mémoire. Ceci signifie que tous les mouvements nécessaires pour dessiner une marque sont structurés en une seule unité avant le début du geste et sont exécutés en un mouvement fluide [Kurtenbach 93a]. L'exécution de la commande est donc perçue comme une seule action, le dessin d'une marque, et non pas comme une succession de gestes orientés, correspondant chacun à un sous-menu. Les Compound Marking menus permettent donc un *accès direct* aux commandes.

Pour les *Multi-Stroke menus* [Zhao 04], l'utilisateur ne fait plus une seule marque composée, mais une série de marques simples (Figure 24). L'association commande-geste est probablement moins évidente [Zhao 04] que précédemment car les gestes impliqués nécessitent plusieurs actions. En effet, le fait de devoir appuyer/relâcher plusieurs fois le bouton de la souris ne permet pas de conserver un mode kinesthésique [Sellen 90]. Les gestes des Multi-Stroke menus ne seraient peut-être pas perçus comme un seul mnème, mais plusieurs [Zhao 04] : cette question reste ouverte et nécessiterait des études plus approfondies.

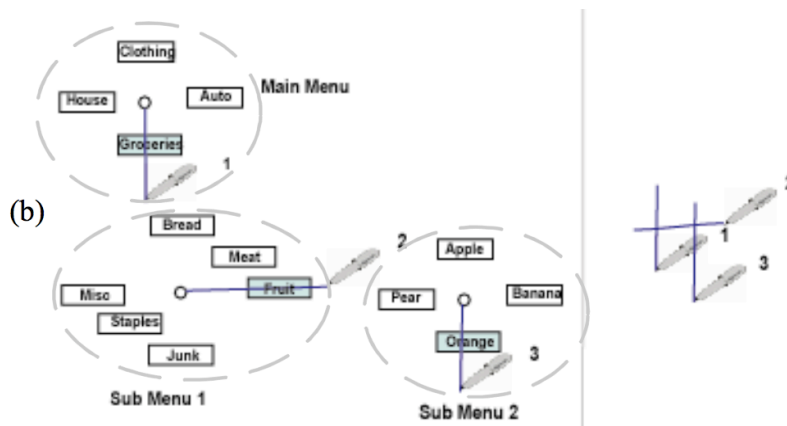
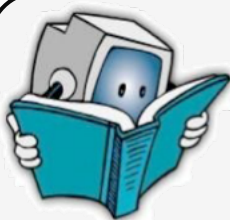


Figure 24 : Les Multi-Stroke menus [Zhao 04]. En mode expert, les utilisateurs dessinent une série de marques simples correspondant au geste pour chaque sous-menu. Illustration extraite de [Zhao 04].

De nombreuses solutions ont été proposées pour améliorer le critère *rapidité et précision*. Elles sont résumées dans le tableau 1 au début de la section. Nous reviendrons sur la navigation dans les systèmes de menus dans le chapitre 6 et sur les groupements internes dans le chapitre 7.



Nous retenons que le critère *rapidité et précision* a suscité beaucoup d'intérêt. Cependant quelques éléments ont été moins étudiés : les solutions pour la navigation en mode novice (chapitre 6) et les groupements internes (chapitre 7).

Nous exposons maintenant le second critère d'utilisabilité : l'apprentissage et la mémorisation.

2 Apprentissage et mémorisation

Une technique de menus doit permettre à l'utilisateur de l'utiliser de manière optimale rapidement et durablement et si possible, d'activer les commandes fréquentes en mode expert. Pour cette raison, nous étudions les techniques de menus au regard du critère *apprentissage et mémorisation*.

		Aspects	Solutions
Facteur d'utilisabilité	Critère apprentissage et mémorisation	Technique	« pointer cliquer » ajout de modalités équivalentes retour visuel affordance
		Organisation	Groupement hiérarchique Groupement interne Vocabulaire Stabilité à travers le temps Stabilité entre applications Adaptabilité Prévisualisation Visibilité du chemin Orientation
		Mode expert	Mise en évidence des relations entre le geste et la commande Ajout de modalités redondantes en sortie Coût Transition fluide

Tableau 3 : Tableau récapitulatif des principales solutions existantes pour le critère *apprentissage et mémorisation*.

On peut distinguer trois phases en ce qui concerne l'apprentissage et la mémorisation du système de menus d'une application. Tout d'abord, si la technique est nouvelle pour l'utilisateur, il doit apprendre son fonctionnement (*apprentissage de la technique*). Avec son expérience, il mémorise ensuite l'organisation des commandes (*apprentissage et mémorisation de l'organisation*). Enfin, vient l'apprentissage du mode expert des commandes utilisées fréquemment en essayant de les retenir même après une période d'inactivité (*apprentissage et mémorisation du mode expert*).

Ces trois phases constituent les différents aspects que nous étudions plus en détail ci-après. Au préalable, nous présentons des faits importants sur la mémoire.

2.1 Mémoire

Mémoire à court terme et à long terme

Les études distinguent souvent deux types de mémoire [Atkinson 68, Card 83, Dix 03] : la *mémoire à court terme* et la *mémoire à long terme*.

La mémoire à court terme ou « mémoire de travail » maintient les informations dont l'utilisateur a besoin au moment d'accomplir sa tâche. Elle a des capacités limites autour de 7 +/- 2 mnèmes (« chunk »)[Miller 56] quel que soit l'individu et le travail intellectuel qu'il a pu faire. Cependant les mnèmes ont des granularités différentes selon les compétences de l'utilisateur. Au contraire, la mémoire à long terme permet de se souvenir de faits vécus plusieurs heures, jours, années auparavant. Elle a une capacité de stockage beaucoup plus importante.

Afin que les informations de la mémoire à court terme puissent être *entreposées* (« Storage ») dans la mémoire à long terme, elles doivent d'abord être *encodées* (« encoding »). L'encodage consiste à transformer ces informations en une représentation mentale compatible avec l'entreposage dans la mémoire à long terme. Le passage de la mémoire à court terme à la mémoire à long terme s'appelle la *consolidation* (« consolidation »). Celui-ci s'opère si [Atkinson 68]:

- l'opération est répétée ;
- l'information a une valeur affective importante pour le sujet.
- l'information nouvelle est associée à une autre déjà connue « codage sémantique » ;

Ces trois observations favorisent l'acquisition et la rétention d'informations dans la mémoire à long terme et peuvent être exploitées pour améliorer la mémorisation.

La *répétition* est une stratégie connue et utilisée depuis longtemps : "la quantité de choses apprises dépend du temps passé à les apprendre" [Dix 03]. De plus, les performances sont meilleures lorsque l'apprentissage est étalé dans le temps (« Distribution of practice effect ») [Baddeley 78]. Certaines solutions de l'espace MenUA s'appuient sur la répétition pour améliorer le critère *apprentissage et mémorisation*.

La seconde observation semble moins pertinente pour notre étude, les menus ayant rarement une valeur *affective* importante pour l'utilisateur.

Enfin, plusieurs stratégies basées sur l'*association* sont détaillées ci-après. Bien que rarement évoquée dans la littérature IHM, l'association est sans doute un élément primordial pour assurer l'efficacité des techniques d'interaction en ce qui concerne la mémorisation.

Stratégies pour améliorer la mémorisation

Nous présentons 7 stratégies, en relation avec la stratégie d'*association*, qui améliore la mémorisation [Intelegem].

- 1) **Associations entre les éléments.** Apprendre une grande quantité d'informations déconnectée et non organisée est une tâche difficile. Au contraire, lorsque celle-ci est organisée et porteuse de sens, il est à la fois plus facile de l'*entreposer* et de la *recupérer* en mémoire. La stratégie consiste à expliciter un lien entre les éléments en les rassemblant et en les organisant.
- 2) **Associations avec l'environnement.** Lorsque nous apprenons, nous faisons consciemment ou inconsciemment des associations avec l'environnement (un lieu, un état mental, une pensée du moment). Une stratégie pour récupérer en mémoire un élément appris consiste à se rappeler du contexte. Par exemple, lorsque nous perdons nos clefs, nous essayons de retracer le chemin qui nous a amené à les déposer à un endroit inhabituel.
- 3) **Associations avec des éléments connus.** Pour favoriser l'entreposage d'informations en mémoire et leur récupération, il est possible de coupler l'information à apprendre avec quelque chose de familier ou de connu. Par exemple, il est possible de se rappeler le nom de « Robert Green » en l'imaginant en train de jouer au golf (sur un green).
- 4) **Approche en entonnoir** (« Funnel approach »). Elle consiste à apprendre les concepts généraux avant d'apprendre les détails spécifiques. Lorsque les concepts généraux sont appris en premier, les détails font sens plus rapidement. Les détails ne sont plus déconnectés entre eux, mais rattachés par un cadre plus général. Cette approche consiste, par exemple, à se familiariser avec la table des matières d'un chapitre afin d'avoir une vue d'ensemble avant d'étudier les détails contenus dans le chapitre.
- 5) **Mémoire visuelle.** L'exploitation de la mémoire visuelle peut aider la mémorisation. Ce point est à mettre relation avec les points 2 (associations avec l'environnement) et 3 (associations avec des éléments connus). Par exemple, visualiser une idée en la dessinant

sur une feuille peut favoriser sa mémorisation : la personne associe l'idée à une image. « L'art de mémoire » est une méthode mnémotechnique, pratiquée depuis l'antiquité, basée sur ce principe [Yates 66]. Elle consiste d'abord à associer des éléments nouveaux à apprendre à des lieux déjà bien connus, tels qu'une cathédrale, puis à visiter l'édifice dans l'ordre habituel afin de se remémorer les éléments grâce à la mémoire spatiale .

- 6) **Mémoire auditive.** Une approche complémentaire consiste à réciter à haute voix les informations à apprendre afin d'exploiter la mémoire auditive. L'élément à apprendre n'est plus associé à une image mais à un son.
- 7) **Enseignement des éléments.** S'imaginer en train d'enseigner les éléments à apprendre à un auditoire force à structurer les idées, c'est-à-dire à trouver un lien logique entre les idées, afin qu'elle fasse sens à la fois pour les apprenants et pour soi-même.



Nous retenons que la répétition et les 7 types d'association peuvent avoir un impact sur le critère *apprentissage et mémorisation*.

Nous présentons maintenant, en fonction de ce qui précède, différentes solutions organisées selon les trois aspects considérés :

- apprentissage de la technique ;
- apprentissage et mémorisation de l'organisation ;
- apprentissage du mode expert.

2.2 Apprentissage de la technique

Ce critère peut être mis en relation avec ceux de « predictability », « synthesizability » [Abowd 92] ou celui « d'incitation » [Bastien 93]. Une nouvelle technique doit être suffisamment simple à prendre en main sinon elle risque de ne pas être utilisée. Le fonctionnement d'une technique doit s'approcher du modèle mental que s'en fait l'utilisateur afin que celui-ci puisse s'en servir rapidement et sans effort.

Les menus linéaires sont de loin les menus les plus utilisés dans les logiciels courants³. Ils reposent sur une interaction de type *pointer et cliquer* qui constitue généralement la seule référence connue de l'utilisateur. Lorsqu'une nouvelle technique ne repose pas sur ce mode, il est préférable qu'elle dispose également d'un mode d'interaction "équivalent" basé sur pointer et cliquer afin de ne pas désorienter l'utilisateur [Coutaz 94]. Par exemple, les menus circulaires ne permettant pas d'interagir en cliquant risquent de décourager l'utilisateur même si la technique est efficace à long terme.

Une autre solution consiste à améliorer l'affordance de la technique afin de suggérer son fonctionnement à l'utilisateur. L'affordance [Gibson 79, Gibson 82], et plus précisément l'affordance perçue [Norman 99], est "l'information provenant des caractéristiques physiques du système, perçue par l'utilisateur, influençant la manière dont il utilise ce système". Les retours visuels (« feedback ») peuvent aider à comprendre le comportement de la technique. Par

³ Nous pouvons trouver quelques Split menus dans les applications de bureautique, Pie menus dans certains jeux et Marking menus dans le logiciel Maya.

exemple, les Marking menus classiques fonctionnent sur appui-déplacement-relâchement (« press-drag-release »). Un utilisateur pensant que ces menus s'ouvrent sur un clic souris (comme ceux de Windows) risque de n'avoir aucun retour visuel sur cette action, le menu s'affichant uniquement après un délai. Pour cette raison, Maya [Maya] affiche très brièvement le menu lorsque l'utilisateur relâche le bouton de la souris. Ce retour visuel indique la présence d'un menu et suggère un appui prolongé.



Nous retenons que la prise en main d'une nouvelle technique peut être améliorée en ajoutant des modalités équivalentes ou en utilisant des retours visuels suggèrent le fonctionnement de la technique.

2.3 Apprentissage et mémorisation de l'organisation

L'aspect *recherche visuelle* (section 1.2) consistait à étudier comment l'utilisateur trouvait/identifiait un élément dans le système de menus. Nous étudions ici comment l'utilisateur mémorise l'emplacement de cet élément. Cette section partage donc des similitudes avec la section sur la recherche visuelle.

2.3.1 Structure du menu

Une hiérarchie de commandes qui a du sens facilite l'apprentissage et la mémorisation de ses commandes [Shneiderman 04, Intelegem]. Elle peut être décrite par sa largeur (le nombre maximum de commandes à un niveau) et sa profondeur (le nombre maximum de niveaux dans le système de menu). Pour un nombre de commandes donné, il est donc possible de les organiser plutôt en largeur (faible profondeur) ou en profondeur (faible largeur). Ce balancement est discuté dans le chapitre 7. D'après [humanf. 03], il est préférable pour le critère *apprentissage et mémorisation* d'avoir une structure en largeur qui aide la construction d'une représentation mentale de la hiérarchie. Cette structure favorise une vue globale, utile à la mémorisation (cf. *approche en entonnoir* [Intelegem]).

Les groupements hiérarchiques et internes (logiques) organisent sémantiquement les éléments et mettent en évidence les "associations entre les éléments" favorisant leur récupération en mémoire [Intelegem]. Nous reviendrons sur l'importance des groupements internes dans le chapitre 7.

2.3.2 Vocabulaire

Des items porteurs de sens (*association entre éléments*) grâce à un vocabulaire compréhensible et cohérent guident l'utilisateur et encouragent l'apprentissage de l'organisation [Norman 91, Lee 93, Intelegem, Scapin 86].

2.3.3 Stabilité dans le temps

Une organisation stable favorise son assimilation. Les utilisateurs ne peuvent pas apprendre la position des items si ces derniers sont constamment déplacés surtout si ce déplacement est effectué par le système à l'insu de l'utilisateur. Certains menus comme les *Frequency-ordered menus* [Lee 04] ou les *Folded menus* (présentés dans la section 1.2.2) ne préservent pas la stabilité spatiale car les items sont dynamiquement déplacés. Le menu « Start » de Windows, pour lequel des programmes sont régulièrement ajoutés ou supprimés par l'utilisateur, présente le même inconvénient. Ce problème a été récemment résolu dans Vista

[VistaStartMenu] où l'espace dédié aux items supprimés reste vide afin d'éviter les changements incessants de la disposition spatiale .

La stabilité dans le temps améliore l'apprentissage car elle exploite la *mémoire visuelle* de l'utilisateur et la stratégie de *répétition* [Findlater 04, Intelegen].

2.3.4 Stabilité entre applications

Une organisation est *stable entre les applications*, si elle est semblable dans plusieurs applications. Cette propriété peut être rattachée au critère de « familiarité » ou de « généralisation » dans [Abowd 92]. Elle favorise l'apprentissage par transposition (ou transfert) des connaissances (*associations avec des éléments connus*) [Intelegen]. Un utilisateur peut apprendre rapidement une nouvelle organisation même s'il ne l'a jamais utilisée préalablement en identifiant les différences et les similitudes avec celles qu'il connaît déjà.

Les menus attachés à la barre de menus respectent généralement cette propriété du moins en ce qui concerne les menus « File » et « Edit ». Au contraire, il n'existe pas à notre connaissance, un vocabulaire gestuel standardisé pour les Marking menus. Ceci pourrait certainement aider à la diffusion de ce type de menus dans les applications grand public en évitant aux utilisateurs d'apprendre à chaque fois de nouveaux gestes.

2.3.5 Personnalisation

Une technique de menus est adaptable lorsque l'utilisateur peut la personnaliser. Cette solution favorise l'assimilation de l'organisation en impliquant l'utilisateur dans le processus d'apprentissage (*apprentissage actif* [Intelegen]). Dans [Kurtenbach 99, Findlater 04], l'utilisateur participe activement au positionnement des items. Cette flexibilité est appréciée des utilisateurs [Findlater 04]. Cependant, Findlater n'observe pas de différences significatives de performances par rapport aux menus statiques [Findlater 04].

2.3.6 Navigation

Le fait de naviguer dans un système de menus favorise l'apprentissage de son organisation [Intelegen]. D'une part, l'utilisateur peut se construire plus facilement une représentation mentale de la hiérarchie (*approche en entonnoir*) et d'autre part, il est impliqué dans le processus d'apprentissage (*apprentissage actif*). Ainsi, en facilitant la navigation (section 1.2.5), la prévisualisation ou la visibilité du chemin, facilitent également l'apprentissage et la mémorisation de l'organisation (un point sur lequel nous reviendrons au chapitre 6 consacré à la navigation).

2.3.7 Orientation

En utilisant l'orientation pour organiser les items, les menus circulaires exploitent la « mémoire visuelle » de l'utilisateur [Intelegen]. L'utilisateur peut facilement associer une orientation à une commande et s'en rappeler.

L'orientation favorise les associations entre les éléments en permettant les oppositions sémantiques. Par exemple, les commandes "ouvrir" et "fermer", opposées sémantiquement peuvent être placées dans des orientations opposées. Une fois l'orientation de la commande "Ouvrir" apprise, l'utilisateur peut en déduire facilement l'orientation de la commande « Fermer » [Soliz 03] (*associations avec des éléments connus* [Intelegen]). La mise en évidence de l'opposition sémantique entre les commandes grâce à l'orientation favorise l'apprentissage de l'organisation [Soliz 03].

De manière générale, l'ajout de dimensions (orientation, profondeur, courbure, etc.) offre davantage de possibilités aux concepteurs pour organiser les éléments. Il est alors plus facile de mettre en avant des associations entre les éléments. Nous reviendrons sur cette solution dans le chapitre 7.



Nous retenons que l'apprentissage de l'organisation peut être facilité par une structure compréhensible, cohérente et stable. Il peut également être favorisé lorsque la technique incite à la navigation. Enfin l'ajout de dimensions, telles que l'orientation, favorise la création d'associations entre les éléments.

Nous reviendrons sur cette dernière solution dans le chapitre 7.

2.4 Apprentissage et mémorisation du mode expert

Le mode expert nécessite une phase d'apprentissage contrairement au mode novice qui exploite la reconnaissance. Nous considérons ci-après les raccourcis claviers des menus linéaires et l'interaction gestuelle des menus circulaires.

2.4.1 Raccourcis claviers

Bien que les raccourcis claviers permettent une interaction rapide, Lane observe dans [Lane 05] que la plupart des utilisateurs, même expérimentés, ne font pas l'effort de les apprendre. Ceci s'explique par le fait qu'il existe un compromis entre l'efficacité à long terme et la productivité à court terme [Carrol 87]. En effet, les utilisateurs ne souhaitent pas être détournés de leur tâche principale, ce qu'imposerait l'apprentissage d'un nouveau raccourci clavier. Ils préfèrent souvent continuer avec le mode d'interaction qu'ils connaissent déjà [Krisler 08].

Pour favoriser l'association avec des éléments connus [Intelegen], il est préférable d'utiliser des raccourcis claviers en relation avec le nom de la commande en utilisant par exemple la première lettre de la commande (Ctrl+O pour la commande « ouvrir »). Malheureusement, comme le nombre de lettres de l'alphabet est limité, ils sont souvent forcés d'utiliser des raccourcis claviers plus complexes sans relation simple avec le nom de la commande (Ctrl+D pour la commande « Font » dans Microsoft Word) ou avec une combinaison complexe de touches (Alt + Shift + CMD + C dans Microsoft Word sur Macintosh).

Afin d'apprendre le mode expert, l'utilisateur doit d'abord le découvrir. En 1995, Dix fait le constat qu'il est impossible d'exécuter une commande à l'aide de son raccourci clavier lorsque le menu est affiché. Dans cette situation, l'utilisateur peut penser que le raccourci ne fonctionne pas et ne plus les utiliser. Depuis, les raccourcis claviers fonctionnent même lorsque le menu est affiché afin de laisser l'utilisateur découvrir ce mode de fonctionnement en l'essayant (« learn by doing » [Dix 95]). Il proposait également d'afficher le raccourci clavier sur les boutons iconiques de la barre d'outils comme pour les items dans les menus. Aujourd'hui, des bulles d'aide indiquent parfois le nom et le raccourci clavier associé au bouton après un certain délai.

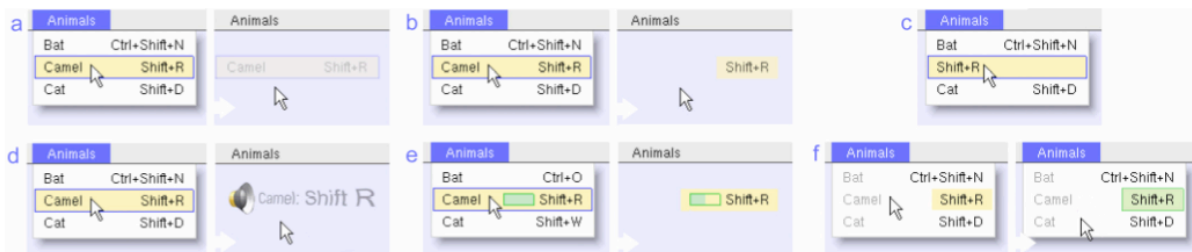


Figure 25 : Plusieurs solutions pour améliorer l'apprentissage des raccourcis claviers. A) conception classique, b) ajout d'un retour visuel après activation, c) remplacement du nom de la commande par le raccourci clavier lorsque le curseur est au dessus de l'item, d) utilisation d'un retour audio, e) introduction d'un délai, f) désactivation de la sélection par le menu. Illustration extraite de [Grossman 07].

Grossman a étudié plusieurs solutions dans [Grossman 07] (Figure 25) pour faciliter l'apprentissage des raccourcis claviers. Deux d'entre elles ont plus particulièrement retenu son attention : le *retour audio* et le *coût*.

La première solution consiste à exploiter « la mémoire auditive » [Intelegen] en augmentant l'activation d'un item d'un retour audio (ajout d'une modalité redondante [Coutaz 94]). Lorsque l'utilisateur clique sur un item, le nom de la commande et son raccourci clavier sont joués par un synthétiseur vocal (Figure 25-d). Le retour audio peut néanmoins gêner l'utilisateur dans un espace public [Buxton 89].

La seconde solution consiste à introduire un coût (« cost-approach »). Les commandes peuvent être vues dans le menu, mais pas activées (Figure 25-f). L'utilisateur est forcé d'utiliser les raccourcis claviers et de les apprendre par la « répétition ». Cette solution peut frustrer un utilisateur inexpérimenté qui risque de rejeter la technique.

Une autre solution consiste à guider les utilisateurs en augmentant les claviers [Logickeyboard, Optimus]. Par exemple, les claviers Logickeyboard [Logickeyboard] et Optimus [Optimus], illustrés Figure 26, affichent l'icône de la commande sur la touche du raccourci clavier correspondant (physiquement pour le Logickeyboard et à l'aide d'écrans miniaturisés pour l'Optimus). Ainsi, les utilisateurs peuvent apprendre la position des raccourcis claviers en les exécutant d'abord à l'aide de leur icône.



Figure 26 : A gauche, le clavier Logickeyboard adapté pour le logiciel Final Cut [Logickeyboard]. A droite, le clavier Optimus [Optimus] où chaque touche est un petit écran.

Malgré la popularité des menus linéaires, et la sous-exploitation de leurs raccourcis clavier [Lan 05], on trouve relativement peu d'études consacrées à ce problème. La plupart des travaux de recherche concernent surtout l'amélioration des performances temporelles en mode expert.

Nous étudions maintenant l'apprentissage du mode expert lorsque celui-ci est basé sur l'interaction gestuelle.

2.4.2 Interaction gestuelle

Un geste s'avère généralement plus facile à mémoriser qu'une combinaison de touche car l'apprentissage d'un geste exploite la *mémoire spatiale* de l'utilisateur [Intelegen] : un geste est généralement représenté par une marque.

Comme pour les raccourcis claviers, il est possible de mettre en relation une commande et un geste (« associations avec des éléments connus »), par exemple, en associant le geste « droit » à la commande « Next » dans un navigateur Web. Dans [Zhao 7b], des marques d'un Multi-Stroke menu ont été utilisés pour dessiner des symboles chinois. Ainsi, le mot « Gros » en chinois correspond aux trois marques de la Figure 27. Zhao montre que les utilisateurs d'origine chinoise apprennent ces gestes plus facilement.

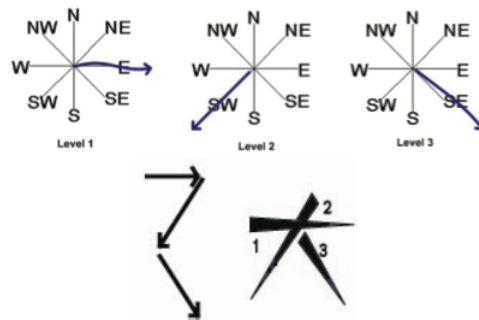


Figure 27 : Construction visuelle d'un symbole chinois à partir de trois marques d'un Multi-Stroke menu. Illustration extraite de [Zhao 07b].

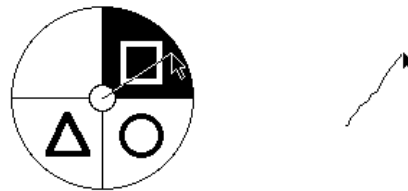


Figure 28 : Transition fluide du mode novice au mode expert pour les Marking menus [Kurtenbach 91] : les gestes sont identiques en mode novice (à gauche) et en mode expert (à droite). Illustration extraite de [Kurtenbach 91].

Les Marking menus [Kurtenbach 91] utilisent les mêmes gestes en mode novice et en mode expert. La seule différence entre ces deux modes de fonctionnement est que le menu n'est pas affiché en mode expert (Figure 28). Les utilisateurs apprennent donc le mode expert implicitement, juste en « répétant » les gestes en mode novice. Les gestes sont encodés progressivement dans la mémoire musculaire de l'utilisateur et le retour visuel devient inutile. Les Marking menus permettent ainsi une *transition fluide* entre ces deux modes de fonctionnement.

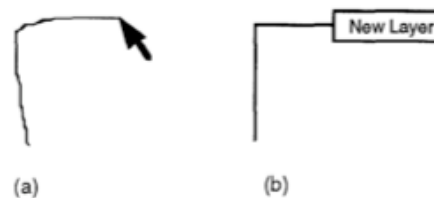


Figure 29 : Après l'activation d'une commande avec les Marking menus, la marque parfaite est dessinée quelques dixièmes de secondes. Illustration extraite de [Tapia 95].

Une technique pour favoriser l'apprentissage des gestes consiste à afficher la marque parfaite quelques dixièmes de secondes après l'activation d'une commande. L'utilisateur voit la marque parfaite qui correspond à la commande qu'il vient d'activer et maintient aussi son attention sur le mode expert [Tapia 95, Grossman 07].



Nous retenons que l'apprentissage du mode expert peut être favorisé par des techniques, telles que les Marking menus, qui exploitent l'interaction gestuelle.

Notre contribution, dans cette section, est d'avoir explicité des solutions, de les avoir expliquées au regard des stratégies améliorant la mémorisation et de les avoir mises en relation avec les différents aspects du critère *apprentissage et mémorisation*.

Ce critère a été nettement moins étudié que celui de *rapidité et précision*. Ceci s'explique peut-être par l'absence de lois empiriques connues. Le modèle de la mémoire [Akinston 68], présenté en début de section, a l'avantage d'être suffisamment simple pour extraire quelques éléments d'étude pertinents. Cependant, il ne traduit pas la complexité de la mémoire et du fonctionnement du cerveau humain.

Le critère *apprentissage et mémorisation* nous semble essentiel pour l'acceptation d'une technique de menus. Pour cette raison, nous proposons, dans le chapitre 7, une nouvelle technique de menus qui vise à améliorer ce critère.



Nous retenons que le critère *apprentissage et mémorisation* a été étudié nettement moins que le critère *rapidité et précision* alors qu'il nous apparaît comme étant au moins aussi important pour l'acceptation d'une technique de menu.

Nous revenons sur ce critère dans le chapitre 7.

Nous étudions maintenant le dernier critère d'utilisabilité, la satisfaction, qui est à la frontière de notre étude.

3 Satisfaction

		Aspects	Solutions
Facteur d'utilisabilité	Critère satisfaction	Viscéral	Principes gestalts (simplicité, symétrie, fermeture, proximité, bonne forme, centre de gravité, transportabilité)
		Comportemental	Prise en main Interaction gestuelle
		Réflexif	

Tableau 4 : Tableau récapitulatif des principales solutions existantes pour le critère *satisfaction*.

La satisfaction est un critère complexe qu'il est difficile de définir précisément [Lindgaard 07]. Il peut être rattaché à d'autres termes comme l'affectif, l'émotion, l'esthétisme, l'attirance visuelle, l'expérience utilisateur, qui sont tous subjectifs. A notre connaissance, il n'existe aucune étude focalisée sur la satisfaction en ce qui concerne les techniques de menus. Aussi nous étudions les définitions de la satisfaction en général, que nous instancions au cas d'une technique de menu.

3.1 Expérience utilisateur

Selon Shneiderman [Shneiderman 04], la satisfaction est un critère associé à l'utilisabilité, au même titre que la rapidité, la précision, l'apprentissage ou la mémorisation. La satisfaction est donc vue comme un critère permettant d'améliorer l'utilisabilité. A l'opposé, des travaux plus récents [Norman 04, Buxton 05, Law 08] affirment que l'utilisabilité, comme l'utilité [Nielsen 93], est une composante du critère de satisfaction. La satisfaction est alors le point culminant de *l'expérience de l'utilisateur* [Buxton 05].

Pour Law [Law 08], l'expérience utilisateur dépasse les limites du traditionnel cadre de l'utilisabilité (aspect cognitif et aspect de performance de l'utilisateur) et de l'utilité en prenant en compte d'autres aspects tels que l'interaction « non-utilitaire » (non-utilitarian) comme l'affect et la sensation. Les attributs qui interviennent dans l'expérience utilisateur incluent : l'amusement, le plaisir, la fierté, la surprise et l'intimité.

Dans cette thèse, nous adhérons à l'approche de Shneiderman en considérant la satisfaction comme critère d'utilisabilité [Shneiderman 04]. Néanmoins, il convient de noter que d'autres travaux [Law 08] situent l'expérience au-delà de l'utilisabilité. Nous étudions maintenant le modèle de l'émotion de Norman [Norman 04] qui met en évidence trois niveaux utiles pour l'analyse des menus.

3.2 Emotion

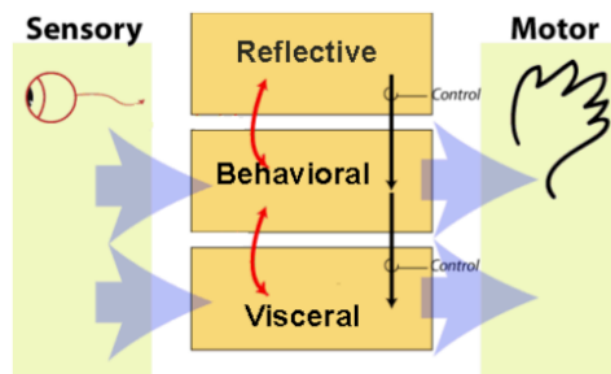


Figure 30 : Le modèle de l'émotion. Illustration extraite de [Norman 04].

Norman [Norman 04] conçoit l'émotion comme un processus de traitement de l'information à trois niveaux : le niveau viscéral (« visceral level »), le niveau comportemental (« behavioral level ») et le niveau réflexif (« reflective level »).

- **Le niveau viscéral** est un processus inconscient et automatique. Il correspond aux premières impressions en permettant des jugements rapides, déterminés biologiquement, sur ce qui est "bon" ou pas. Il fournit une réponse basée entièrement sur les aspects de surface d'un objet. L'apparence à travers la forme, la couleur ou les proportions affecte la préférence de l'utilisateur à ce niveau. Il est important de ne pas négliger le niveau viscéral, car c'est le point de départ du processus affectif qui va conditionner les autres niveaux. Par exemple, Lindgaard montre dans [Lindgaard 06] que les utilisateurs se font une idée d'une page web dans les premières 50 millisecondes. Si la raison demeure importante, tout commence donc par un traitement affectif de l'information
- **Le niveau comportemental** est toujours inconscient mais contrôle les « comportements » humains. Il est associé au plaisir d'utilisation d'un objet. La prise en main de cet objet, sa compréhension, ses performances, ou son utilité agissent à ce niveau. Il correspond à la "beauté" d'une opération.

- **Le niveau réflexif** est le siège du conscient. Il fait appel à la réflexion, aux pensées, aux sentiments et aux émotions. Contrairement aux deux autres niveaux, il n'a pas d'accès direct à l'entrée sensorielle ni au contrôle moteur. C'est une réponse qui va au-delà de l'aspect de surface de l'objet et qui est influencée par les connaissances, la formation et la culture de l'utilisateur. Il porte sur l'image de soi.

Selon Norman, chacun de ces niveaux présente des caractéristiques pour la conception d'un produit :

- l'apparence pour le niveau viscéral (impact immédiat d'un produit) ;
- le plaisir d'utilisation pour le niveau comportemental ;
- l'image de soi, la satisfaction personnelle, la mémoire pour le niveau réflexif (interprétation, compréhension, raisonnement).

L'esthétique peut agir sur ces niveaux. Une partie de l'esthétique correspond au niveau viscéral. Elle fait appel à des notions comme l'attraction, l'apparence, et est une réponse basée entièrement sur les aspects de surface d'un objet. Une autre partie correspond au niveau comportemental : c'est la beauté de l'opération, du comportement. Ces deux premiers niveaux de l'esthétique sont subconscients. Une dernière partie appartient au niveau réflexif, elle fait appel à des notions comme la beauté, la réflexion consciente, l'expérience. C'est une réponse qui va au-delà de l'aspect de surface de l'objet et qui est influencée par les connaissances, la formation et la culture de l'utilisateur.

Nous retenons que la satisfaction est une notion complexe et que les trois niveaux (viscéral, comportemental et réflexif) du modèle de l'émotion [Norman 04] peuvent aider la conception d'un produit et en particulier une technique de menu vis-à-vis du critère *satisfaction*.

3.3 Satisfaction et menus

Comme annoncé, peu de travaux étudient les menus, et plus largement les techniques d'interaction, sous l'angle de la satisfaction. Les utilisateurs sont parfois soumis à des questionnaires à la fin des expériences qui sont centrés sur l'utilisabilité perçue. Nous visons ici, à identifier quelques pistes pour l'étude de la satisfaction des techniques de menus.

Niveau viscéral

Les Marking menus agissent certainement au niveau viscéral par leur forme. Soliz [Soliz 03] montre qu'ils suivent des principes Gestalt [Helson 33] comme :

- La simplicité et la symétrie (« simplicity and symmetry ») : la simplicité de la forme, et particulièrement la symétrie permettent de créer l'idée d'un *tout*. Par exemple, la forme circulaire des Marking menus est à la fois simple et symétrique.
- La fermeture (« Closure ») : le cerveau aime regrouper des formes de façon à constituer des ensembles visuels fermés et cohérents. Dans le cas des Marking menus, il est facile de reconstituer un cercle même lorsqu'il manque des items.
- La proximité (« proximity ») : plus les éléments sont proches les uns des autres, plus ils sont perçus comme ayant un rapport entre eux. Ils forment des sous-ensembles qui vont fonctionner comme un « tout ». Par exemple, l'organisation circulaire des items dans un Marking menus, minimise la distance totale entre les items et tend à renforcer ce *tout*.
- Le centre de gravité (« center of Gravity ») : les formes reposant sur les principes gestalt sont généralement organisées autour d'un centre de gravité. Par exemple, la zone centrale des Marking menus sert de centre de gravité.

D'autres attributs visuels, comme la couleur agissent également au niveau viscéral et peuvent favoriser la satisfaction de l'utilisateur durant les premières expériences. Cependant, il est important de noter que les menus ne sont pas l'objet principal de la tâche. Ils ne doivent donc

pas détourner l'attention des utilisateurs souhaitant rester concentrés sur leur tâche. Nous observons que les menus ont généralement peu de couleurs et sont discrets.

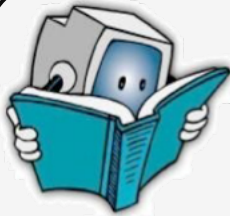
Niveau comportemental

Le critère de satisfaction est certainement favorisé, au niveau comportemental, par une technique facile à prendre en main. Une technique difficile à maîtriser risque d'être abandonnée (section 2.2).

L'interaction gestuelle des menus circulaires transforme une marque en une commande qui peut plaire à l'utilisateur grâce à la *beauté de l'opération* [Norman 04].

Niveau réflexif

L'utilisation d'une nouvelle technique efficace valorise certainement son utilisateur auprès de son entourage. [Lane 05] montre par exemple que nous utilisons davantage les raccourcis claviers si nous appartenons à un groupe qui les utilisent.



Nous retenons les trois niveaux (viscéral, comportemental et réflexif) de [Norman 04] comme point de départ pour l'étude de la satisfaction pour les techniques de menus.

4 Conclusion

Dans ce chapitre, nous avons étudié les techniques de menus et leurs solutions conceptuelles sous l'angle de l'utilisabilité, première facette de notre espace de critères MenUA.

Ce chapitre souligne un déséquilibre sur la répartition des travaux existants en fonction des critères. Il apparaît clairement que le critère *apprentissage et mémorisation* a suscité moins d'intérêt que celui de *rapidité et précision*. Aussi, le chapitre 7 se concentre sur ce critère d'apprentissage qui nous semble essentiel pour l'acceptation d'une technique de menus.

L'étude du critère *rapidité et précision* a permis de mettre en évidence des éléments moins étudiés, comme ceux favorisant la navigation dans les menus. En effet, les travaux se sont principalement concentrés sur les performances des utilisateurs expérimentés qui n'utilisent pas de chemins détournés. Par exemple, la plupart des évaluations expérimentales avec des Marking menus se sont focalisées sur le mode expert [Kurtenbach 93, Balakrishnan 98, Zhao 04, Zhao 06]. Aussi, nous soulignons dans le chapitre 6, l'importance du mode novice et de la prise en compte des chemins détournés pour la conception de techniques de menus. Nous mettons en évidence les solutions associées, et proposons une nouvelle technique qui améliore la navigation des Marking menus en mode novice.

Enfin, nous avons abordé dans ce chapitre le critère *satisfaction*, une notion complexe et peu étudiée. Nous avons considéré les trois niveaux du modèle de l'émotion de Norman [Norman 04]. Ce modèle peut fournir un point de départ pour une analyse plus systématique des techniques de menus et de leurs solutions vis à vis de ce critère. L'étude du critère de satisfaction apparaît comme une voie prometteuse, mais difficile, qui invite à adopter un nouveau regard sur les techniques de menus (et plus largement sur les techniques d'interaction).

Nous présentons dans le chapitre suivant, la seconde facette de l'espace MenUA, le facteur d'applicabilité.

Applicabilité des techniques de menus

Chapitre 5



1	ADEQUATION A L'APPLICATION	84
1.1	NOMBRE D'ITEMS EN LARGEUR	84
1.1.1	MENUS CIRCULAIRES	85
1.1.2	MENUS LINEAIRES	87
1.2	PROFONDEUR DU SYSTEME DE MENUS	89
1.3	NOMBRE TOTAL D'ITEMS	90
2	ADEQUATION A LA PLATEFORME	92
2.1	DISPOSITIFS D'ENTREE	92
2.1.1	ABSENCE DE BOUTON DROIT	92
2.1.2	ABSENCE DE CLAVIER	93
2.1.3	OCCULTATION	93
2.1.4	IMPRECISION DE L'INTERACTION AU DOIGT	93
2.1.5	OPPORTUNITES : SPECIFICITES DE CERTAINS DISPOSITIFS	95
2.2	CONSTRAINTES SPATIALES	95
3	ADEQUATION A LA TACHE	97
3.1	CONTEXTE VISUEL	97
3.2	TRANSITIONS	99
3.2.1	SELECTION DU MENU (TRANSITION OBJET-COMMANDE)	99
3.2.2	RETOUR A L'OBJET D'INTERET (TRANSITION COMMANDE-OBJET)	100
3.2.3	RETOUR AU MENU (TRANSITION COMMANDE-COMMANDE)	101
3.2.4	CONTROLE DE VALEURS (TRANSITION COMMANDE-VALEUR)	102
4	CONCLUSION	104

Ce chapitre est consacré au facteur *applicabilité*, le second volet de l'espace MenUA. Nous étudions ici l'adéquation d'une technique de menus par rapport aux besoins applicatifs de l'utilisateur. Le facteur d'*applicabilité* se distingue de l'*utilisabilité* par le fait qu'il évalue les menus dans leur contexte d'utilisation : le menu est-il applicable pour une application et une plateforme donnée ?

Facteur	Critères	Aspects
Applicabilité	Adéquation à l'application	Nombre d'items en largeur
		Profondeur du système de menus
		Nombre total d'items
	Adéquation à la plateforme	Dispositif d'entrée
		Dispositif de sortie
	Adéquation à la tâche	Contexte visuel
		Transitions

Tableau 1 : Facteur d'applicabilité

La section 1 est dédiée au critère *d'adéquation à l'application*. Nous étudions la capacité d'une technique de menus à contenir les commandes de l'application, c'est-à-dire sa capacité à permettre les différentes structures hiérarchiques souhaitées par les concepteurs d'applications.

La section 2 décrit le critère *d'adéquation à la plateforme*. Nous étudions la compatibilité d'une technique de menus donnée avec divers dispositifs d'entrée/sortie.

Enfin, la section 3 traite du critère *d'adéquation à la tâche*. Nous étudions la capacité du menu à s'intégrer dans l'activité de l'utilisateur.

1 Adéquation à l'application

		Aspects	Solutions
Facteur d'applicabilité	Critère d'adéquation à l'application	Nombre d'items en largeur	Vocabulaire gestuel (orientation, position) Barre de défilement Représentation Fish-eye
		Profondeur du système de menus	Composition temporelle de marques
		Nombre total d'items	Composition de menus circulaires et menus linéaires

Tableau 2 : Tableau récapitulatif des principales solutions existantes pour le critère d'adéquation à l'application.

Un système de menus doit permettre d'accéder à l'ensemble des commandes d'une application. Ces dernières peuvent contenir un très grand nombre de commandes. Par exemple, Microsoft Word 6 comprenait 642 commandes en 1999 [Linton 99] et ce nombre ne cesse d'augmenter (maintenant plus de 1000 commandes). Ceci nécessite d'organiser les commandes hiérarchiquement. Une organisation arborescente se caractérise principalement par :

- sa *largeur*, le nombre maximum d'items à un niveau de l'arborescence ;
- sa *profondeur*, le nombre maximum de niveaux dans la hiérarchie.

Le question posée aux concepteurs d'application est celle du choix relatif de ces deux paramètres pour optimiser l'interaction (section 1.3 du chapitre 4-Critère *rapidité et précision*) et l'apprentissage du contenu du menu (section 2.3 du chapitre 4-critère *apprentissage et mémorisation*). Nous revenons sur ce point au chapitre 7. De plus, la capacité de passage à l'échelle (« Scalability » [Beaudouin-Lafon 04]), c'est-à-dire, pour un menu, la possibilité de contenir un nombre suffisant de commandes est un critère applicatif important. Certains types de menus imposent des restrictions drastiques sur le nombre d'items que le menu peut contenir en largeur. Ce manque de flexibilité interdit aux concepteurs d'organiser la structure des systèmes de menus comme ils le souhaitent.

Les trois aspects devant être pris en considération sont :

- le nombre d'items en largeur ;
- la profondeur du système de menus ;
- le nombre total d'items.

1.1 Nombre d'items en largeur

La largeur d'un menu est le nombre maximal d'items qu'il peut contenir à un même niveau de profondeur. On observe que les menus des applications usuelles contiennent souvent 12 commandes ou plus car il est généralement préférable d'adopter des structures en *largeur* plutôt qu'en *profondeur* (chapitre 7, section 1). Une limitation du nombre d'items en largeur impose de séparer arbitrairement des groupes de commandes sémantiquement liées. La structure du système de menus risque alors de ne pas correspondre à la structure mentale de l'utilisateur [Zhao 06] ce qui peut avoir un effet négatif sur l'apprentissage et la mémorisation (section 3.2 du chapitre 4-critère *apprentissage et mémorisation*).

1.1.1 Menus circulaires

La limitation précédente est particulièrement critique pour les menus circulaires. En effet, lorsque le nombre d'items augmente, l'amplitude des secteurs angulaires alloués aux items diminue. Une trop petite taille empêcherait une sélection rapide et précise en mode expert (section 1.5 du chapitre 4). Kurtenbach montre dans [Kurtenbach 93b] qu'il est difficile de maintenir un niveau élevé de précision en mode expert au delà de 8-12 items. Or près de la moitié des menus d'applications grand public contiennent plus de 14 items en largeur (chapitre 7, section 1). Les menus circulaires ne sont donc pas adaptés aux configurations usuelles.

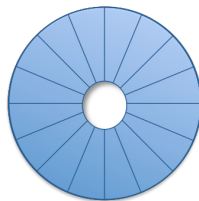


Figure 1 : L'amplitude des secteurs angulaires pour un menu circulaire avec 16 commandes.

Afin d'augmenter la *largeur* des marking menus, plusieurs solutions qui partagent la même approche ont été proposées : considérer de nouvelles dimensions en entrée en plus de l'orientation telles que la distance ou la pression [Nancel 08].

Une solution consiste à exploiter à la fois l'orientation et la position relative des marques par rapport à un point de référence généralement le point d'activation du menu. Les *Zone menus* [Zhao 06], illustrés Figure 2, permettent d'accéder jusqu'à 32 items en largeur. En mode novice, l'utilisateur clique pour activer le menu et spécifier le point de référence. Cette action divise l'écran en 4 zones autour de l'origine. Chaque zone donne accès à un Marking menu avec 8 items en pressant une nouvelle fois le bouton de la souris dans la zone correspondante. $4 \times 8 = 32$ commandes sont donc théoriquement disponibles à chaque niveau. A la Figure 2, nous illustrons les Zone menus dans le cas de 4×4 commandes. En mode expert, l'utilisateur spécifie une origine (par un clic) puis fait une marque dans la zone correspondante. Un inconvénient des Zone menus, est qu'au niveau mental, l'utilisateur a certainement 4 menus de 4 items plutôt qu'un menu de 16 items.

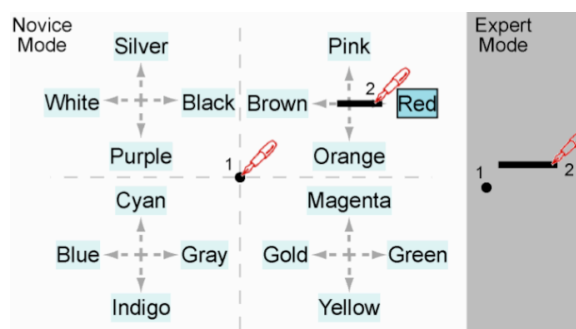


Figure 2 : Les Zone Menus [Zhao 06]. A gauche, le mode novice : l'utilisateur fait un clic (appui - relâchement) pour ouvrir le menu. L'écran est partagé en 4 zones qui donnent chacune accès à un Marking menu de 8 items (ici 4). L'utilisateur doit d'abord se déplacer dans la zone du Marking menu qui l'intéresse puis faire une marque partant du centre de ce menu et en direction de l'item désiré. A droite, le mode expert : L'utilisateur fait la même marque qu'en mode novice (un clic puis une marque relativement au point d'activation). Illustration extraite de [Zhao 06].

Les *Polygon menus* reposent sur le même principe (un clic + une marque) mais les items sont arrangés autour d'un polygone (Figure 3). Cette technique peut accepter jusqu'à 16 items. Un inconvénient des Polygon menus est que l'association entre un geste et sa commande est difficile à apprendre et à mémoriser [Bailly 08] (section 2.4 du chapitre 4). Nous reviendrons sur cet inconvénient dans le chapitre 7.

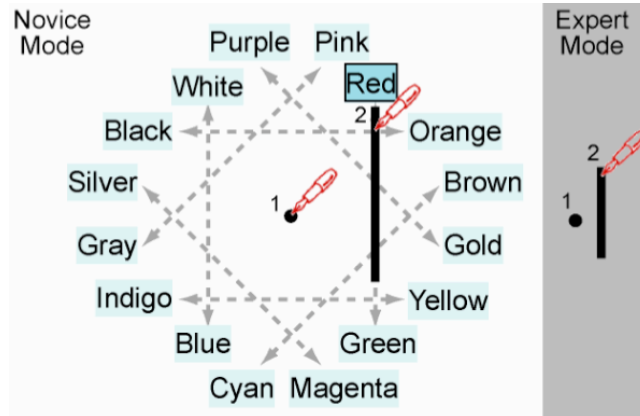


Figure 3 : Les Polygon menus (PM) [Zhao 06]. A gauche le mode novice. A droite, le mode expert. Ces menus fonctionnent comme les Zone menus : l'utilisateur doit faire un clic pour ouvrir le menu puis exécuter une marque en direction de l'item désiré relativement au point d'activation initial. Illustration extraite de [Zhao 06].

Une autre possibilité est d'exploiter la distance en plus de l'orientation [Kurtenbach 93b, Nancel 08]. Les *DartBoard menus*, illustrés Figure 4, permettent d'accéder jusqu'à 32 items en discriminant 4 longueurs de marques pour chacune des 8 orientations.

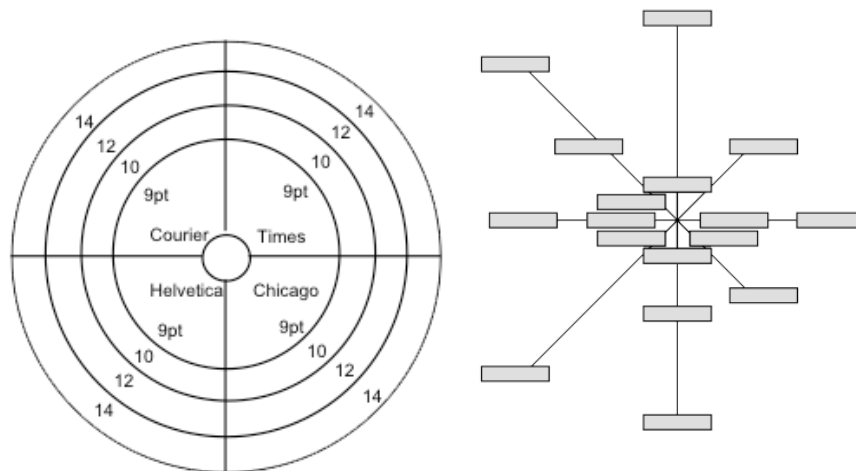


Figure 4 : Les DartBoard menus exploitent à la fois l'orientation et la distance. A gauche, illustration extraite de [Kurtenbach 93b] avec 4*4 items. A droite, illustration extraite de [Nancel 08] avec 19 items.

En fonction du dispositif d'entrée, d'autres dimensions peuvent être utilisées pour les menus comme la pression, le tilt ou les multi-pointeurs [Nancel 08]. Les *Donuts menus* [Lai 05, Ren 08], illustrés Figure 5 ou les Push Menus [Huot 08], illustrés Figure 6, permettent d'accéder à différentes couches (« layers ») d'un Pie menu en fonction du niveau de pression.

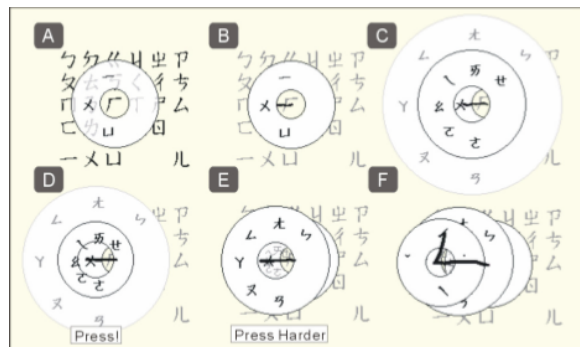


Figure 5 : Les Donuts menus [Lai 05] exploitent la pression pour augmenter le nombre de caractères chinois dans un Pie-menu. Illustration extraite de [Lai 05].

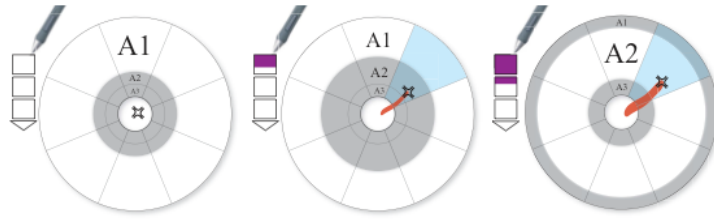


Figure 6 : Les Push Menus [Huot 08] exploitent la pression pour augmenter le nombre d'items en largeur dans un Marking menu. Illustration extraite de [Huot 08].

Dans nos travaux, nous explorons deux nouvelles dimensions pour augmenter le nombre d'items en largeur dans les menus : la courbure avec les *Flower menus* présentés dans le chapitre 7 et le multi-pointeurs avec le *Multi-Touch Menu* dans le chapitre 8.

1.1.2 Menus linéaires

Même si les menus linéaires ne souffrent pas des mêmes contraintes que les menus circulaires, ils peuvent cependant difficilement contenir un très grand nombre d'items (>30) en largeur. Des barres de défilement sont parfois ajoutées lorsque le menu est trop grand pour pouvoir être entièrement affiché à l'écran. Mais ces barres sont encombrantes et ralentissent l'interaction, l'accès aux items demandant souvent une manipulation intermédiaire.

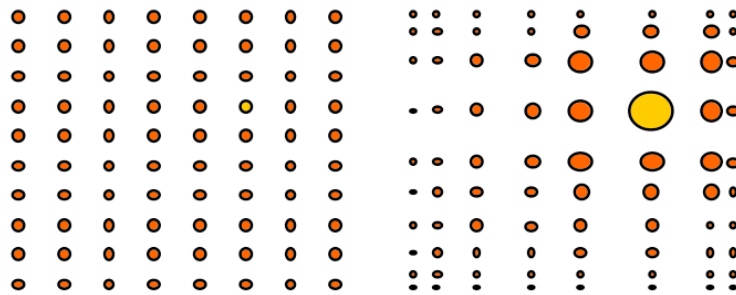


Figure 7 : Une vue en œil de poisson déformante [Furnas 86]. Le résultat (à droite) est obtenu à partir d'une vue initiale (à gauche) et un centre d'intérêt (en plus gros).

Une solution consiste à utiliser une vue en œil de poisson (« Fish eye ») [Furnas 86]. Une vue en œil de poisson modifie la taille des objets en fonction du degré d'intérêt (DOI). Plus le DOI est élevé, plus l'objet est grand et réciproquement, ce qui permet d'afficher un plus grand nombre d'items sur une même surface. Une vue œil de poisson est illustrée Figure 7. Les *Fisheye menus* [Bederson 00], illustrés Figure 8, et leur extension aux menus hiérarchiques, les *FishTree* [Lecolinet 06], illustrés Figure 9, utilisent la vue œil de poisson pour afficher un nombre important d'items simultanément. Les items situés dans la zone d'intérêt¹ sont affichés avec une taille suffisante pour être facilement lisibles. Au contraire, les items situés dans la zone de contexte, sont affichés à une taille inversement proportionnelle à leur distance de la zone d'intérêt.

¹ La taille de la zone d'intérêt peut être changée dynamiquement dans ces deux techniques [Bederson 00].



Figure 8 : Le Fisheye menu [Bederson 00]. Le même menu affiché avec le curseur à trois positions différentes. Illustration extraite de [Bederson 00].

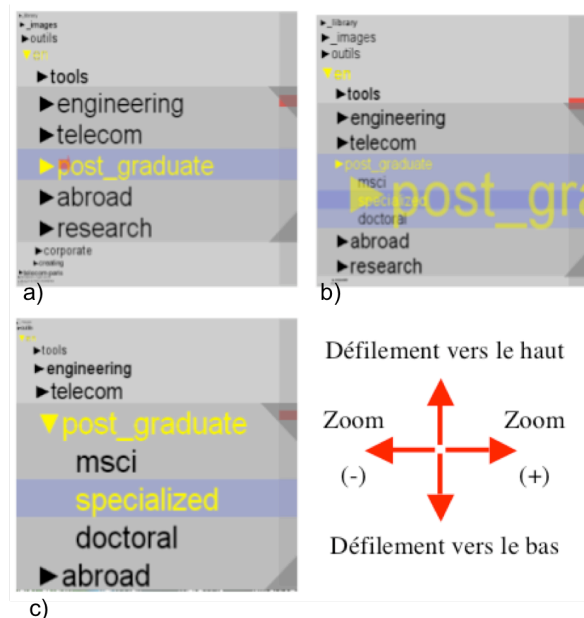


Figure 9 : FishTree menus [Lecolinet 06]. A) l'utilisateur zoome sur l'item « post_graduate ». b) la taille de cet élément grossit et son contenu commence à apparaître en transparence. c) à la fin de l'opération, l'intitulé initial est entièrement remplacé par le contenu du nœud. Illustration extraite de [Lecolinet 06].

1.2 Profondeur du système de menus

Un nombre de commandes important impose d'utiliser des menus hiérarchiques. Ceci est d'autant plus vrai que la largeur du menu est limitée. La profondeur d'un système de menus est le nombre maximal de sous-menus que l'utilisateur peut traverser pour accéder à une commande. Dans cette section, nous nous focalisons sur les menus circulaires qui ont des contraintes plus fortes en profondeur que les menus linéaires.

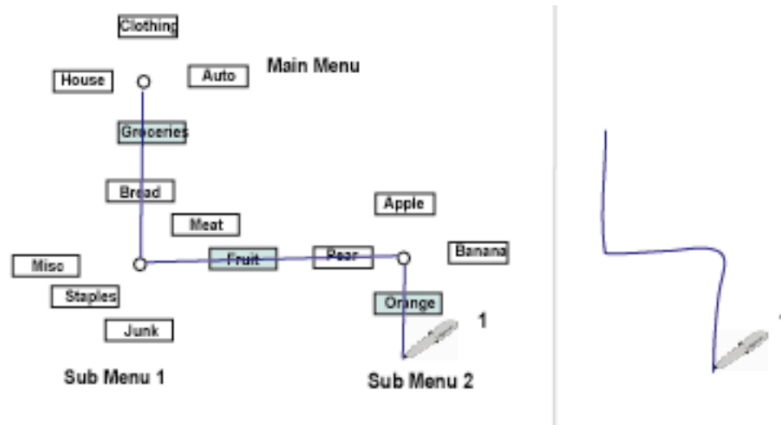


Figure 10 : Un Compound Marking menu de profondeur 3 (présentée dans la section 1.5 du chapitre 4). A gauche le mode novice. A droite, le mode expert : une marque composée de plusieurs inflexions (composition spatiale). Illustration extraite de [Zhao 04]

Les « Compound » Marking menus hiérarchiques, illustrés Figure 10, sont limités en profondeur pour plusieurs raisons. D'abord, les performances chutent avec la profondeur (critère *rapidité et précision* du chapitre 4). Kurtenbach [Kurtenbach 93a] montre qu'il est difficile d'avoir plus de 2 niveaux de profondeur sans dégrader la précision en mode expert. La marque nécessaire pour la sélection d'une commande nécessite une série d'inflexions (zigzag). Cette marque, « composée spatialement », semble difficile à dessiner avec précision en particulier lorsqu'elle implique des gestes diagonaux [Zhao 04]. Le second inconvénient de ces marques est qu'elles sont parfois ambiguës pour une profondeur supérieure à 2. En effet, comme elles sont indépendantes de l'échelle (afin d'augmenter la vitesse de sélection des commandes, section 1.3 du chapitre 4), une même marque peut dans certains cas correspondre à plusieurs commandes. Par exemple, la marque de la Figure 11 pour un Compound Marking menu de profondeur 3 peut correspondre à la commande située en Sud/Sud/Nord-Est ou Sud/Nord-Est/Nord-Est. De manière générale, toutes les commandes qui nécessitent deux gestes consécutifs dans la même direction risquent d'être associées à des marques ambiguës. Une solution simple consiste à interdire certains emplacements dans le menu mais ceci est au détriment du nombre total de commandes. Zhao [Zhao 04] montre que seules 400 sur les $8*8*8=512$ commandes d'un Compound Marking menu de profondeur 3 sont en réalité disponibles en mode expert.

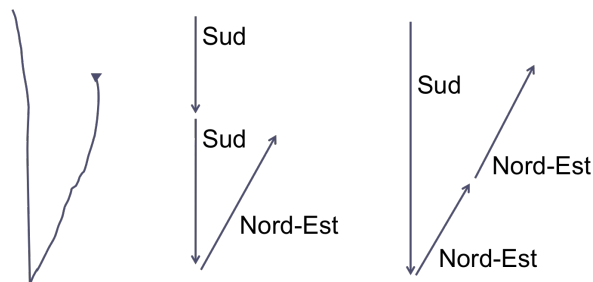


Figure 11 : Ambiguïtés des marques avec les Compound Marking menus.

Une solution pour augmenter la fiabilité des Marking menus hiérarchiques consiste à composer les marques temporellement plutôt que spatialement (section 1.5 du chapitre 4). Nous rappelons que les *Multi-Stroke menus* [Zhao 04] sont des Marking menus qui superposent les sous-menus en mode novice. En mode expert, l'utilisateur dessine une série de marques simples qui peuvent être superposées (Figure 12). Ces marques sont donc composées temporellement (au lieu de spatialement). Le principal intérêt est que ces marques sont indépendantes et garantissent un haut niveau de précision même avec 3 niveaux de profondeur. La précision est peu dépendante du nombre de niveaux comme montré dans [Zhao 04]. De plus, les marques ne sont plus ambiguës. En effet, chaque marque simple (associée à chaque niveau de la hiérarchie) est clairement délimitée de la précédente par le relâchement du bouton de la souris. Un Multi-Stroke menu de profondeur 3 peut donc contenir $8*8*8=512$ commandes. Cependant, en superposant les sous-menus, cette technique dégrade la navigation en mode novice (section 1.2 du chapitre 4) comme nous le verrons au chapitre 6.

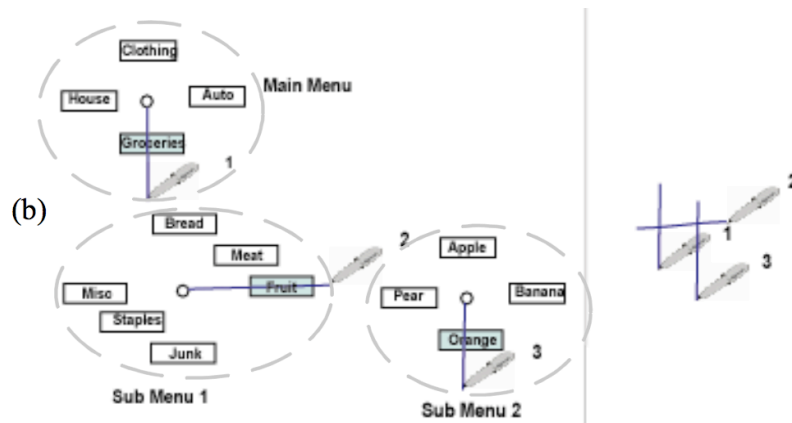


Figure 12 : Les Multi-Stroke menus [Zhao 04] (présentée dans la section 1.5 du chapitre 4). . A gauche, le mode novice. Les sous-menus se superposent. A droite, le mode expert. L'utilisateur fait une série de marques simples qui peuvent être superposées (composition temporelle). Illustration extraite de [Zhao 04]

1.3 Nombre total d'items

Certaines applications contiennent un très grand nombre de commandes (par exemple, plus de 1000 commandes pour MS-Word08). Il ne s'agit plus seulement d'augmenter la largeur ou la profondeur du système de menus, mais les deux dimensions simultanément.

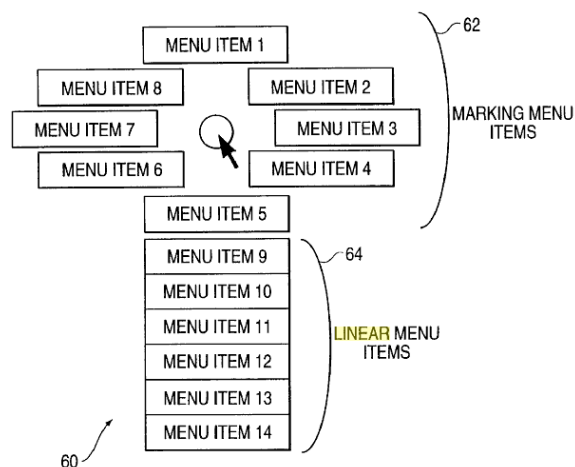


Figure 13 : Menus circulaires avec des portions linéaires [Kurtenbach 97]. Illustration extraite de [Kurtenbach 97].

Afin de contourner les limitations des menus circulaires en ce qui concerne le nombre d'items, une stratégie consiste à les combiner avec des menus linéaires. Kurtenbach [Kurtenbach 97] propose par exemple d'utiliser un menu circulaire au premier niveau donnant accès à des menus linéaires cascadés (Figure 13). Ceci permet d'activer le premier niveau plus rapidement tout en donnant accès un grand nombre d'items.

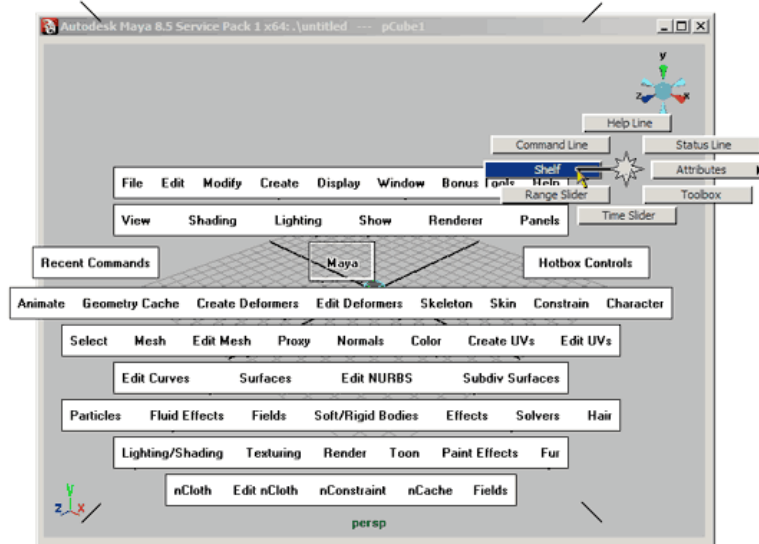
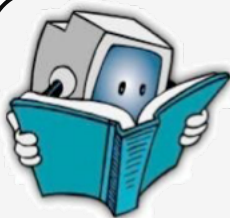


Figure 14 : La Hotbox [Kurtenbach 99]. Lorsque l'utilisateur appuie sur la barre d'espace, des barres de menus s'affichent au centre de l'écran et l'écran est partagé en 4 zones qui donnent accès à différents Marking menus. Illustration extraite de [TutorielMaya].

Finalement, La Hotbox [Kurtenbach 99], illustrée Figure 14, a été conçue pour contenir 1200 commandes. Lorsque l'utilisateur appuie sur la barre d'espace, la Hotbox apparaît et l'écran est divisé en 5 zones. Dans la zone centrale, plusieurs barres de menus sont affichées. Chacune des quatre autres zones périphériques donne accès à un Marking menu hiérarchique. Ces Marking menus peuvent être configurés par les utilisateurs.

Nous retenons de cette section les limites des Marking menus classiques qui peuvent difficilement contenir un nombre important de commandes tant en largeur qu'en profondeur. Des solutions comme les Multi-Stroke menus ou les Polygon menus augmentent le nombre de commandes. La seconde partie de ce manuscrit met en évidence les limites de ces techniques et propose deux améliorations : les *Wave menus* (chapitre 6) et les *Flower menus* (chapitre 7).



Nous retenons que les Marking menus sont limités par le nombre de commandes qu'ils peuvent contenir tant en largeur qu'en profondeur.

Les chapitres 6 et 7 approfondissent ces deux aspects et décrivent chacun une nouvelle technique de menus.

2 Adéquation à la plateforme

		Aspects	Solutions
Facteur d'applicabilité	Critère d'adéquation à la plateforme	Dispositifs d'entrée	Modifieur Bouton physique Délai Double clic Interaction gestuelle Déplacement Augmentation de la largeur des items Projection du curseur Ajout de dimensions (Tilt, pression, etc)
		Contraintes spatiales	Débrayage Superposition

Tableau 3 : Tableau récapitulatif des principales propriétés et stratégies impliquées pour le critère d'adéquation à la plateforme.

Les plateformes matérielles sont de plus en plus variées et les dispositifs d'entrée/sortie hétérogènes. Nous étudions dans cette section la capacité des techniques de menus à rester compatibles avec différents dispositifs et leur capacité à pouvoir être utilisées dans des contextes pour lesquels elles n'ont pas été conçues au départ. Ce critère peut être mis en relation avec la propriété de « reinterpretabilité » dans [Beaudouin-lafon 04]. Nous étudions également la capacité des menus à exploiter les possibilités offertes par certains types de dispositifs.

2.1 Dispositifs d'entrée

L'interaction avec les menus se fait généralement via un dispositif de pointage (2D) relatif² indirect³ comme la souris ou le touchpad. Ils sont parfois utilisés avec un dispositif absolu indirect comme les tablettes graphiques (tablette Wacom) via un stylet. Ce type de dispositif est particulièrement adapté aux menus basés sur les gestes car les utilisateurs sont habitués à manipuler un stylo pour dessiner. Enfin, les menus sont également activables au clavier grâce aux mnémoniques et aux raccourcis claviers (chapitre 2). Ces dispositifs ne sont néanmoins pas toujours disponibles sur toutes les plateformes, entraînant une éventuelle dégradation de l'interaction (comme la précision, section 1 du chapitre 4).

2.1.1 Absence de bouton droit

L'absence de bouton droit sur une souris (ou de son équivalent sur un stylet de tablette graphique) peut être compensé par l'activation d'une touche du clavier (par exemple « Control » sur les Macintosh) lorsque le dispositif en possède un. Dans le cas contraire, (tablet Pc, PDAs et téléphones mobiles, tabletops, tableaux interactifs) plusieurs solutions ont été envisagées :

- un bouton dédié sur le bord de l'écran [Hinckley 06] ;
- le double clic [Wu 03]
- le multi-touch en tapant simultanément avec deux doigts sur les Touchpad des Macintosh ;

² Les périphériques absolus transmettent une position (x, y). Les périphériques relatifs transmettent un déplacement (dx, dy).

³ Un dispositif direct permet un pointage direct sur l'écran. Un dispositif indirect met en jeu une indirection entre l'action sur le périphérique et l'effet sur le curseur

- les Microrolls ("roulement" de pouce) [Roudaut 09] ;
- un délai temporel.

Cette dernière solution, le délai temporel, peut être frustrant pour les utilisateurs expérimentés qui sont amenés à sélectionner beaucoup de commandes en peu de temps et une source d'erreur pour des utilisateurs inexpérimentés.

2.1.2 Absence de clavier

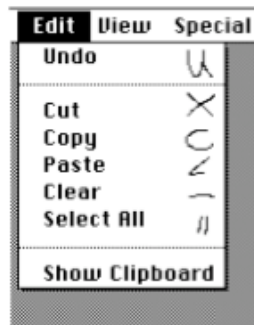


Figure 15 : Menus linéaires augmentés de raccourcis gestuels. Illustration extraite de [Kurtenbach 93b]

L'absence de clavier physique fréquente sur les terminaux mobiles, interdit l'utilisation du mode expert des menus linéaires puisque ceux-ci nécessitent l'activation des raccourcis claviers. Une solution consiste à remplacer les raccourcis claviers par des raccourcis gestuels [Kurtenbach 93b, Bailly 08b, Appert 09]. Nous présenterons au chapitre 8, les Leaf menus qui exploite ce principe.

2.1.3 Occultation

L'occultation est un problème typique des dispositifs directs, en particulier les écrans tactiles des terminaux mobiles, la main ou doigt pouvant masquer une partie des items (Figure 16). Les menus linéaires apparaissent par défaut dans la zone sud-est qui est typiquement la zone recouverte pas la main pour un droitier. L'interaction s'en trouve altérée car l'occultation introduit des mouvements supplémentaires fatigants. Une solution consiste à afficher le menu dans la zone sud-ouest [Hancock 04, Bailly 08b]. Le même problème se pose pour les items situés dans la zone est/sud-est des menus circulaires. Dans le cas des Marking menus, la solution précédente, qui consistait à décaler le menu, semble difficilement envisageable.

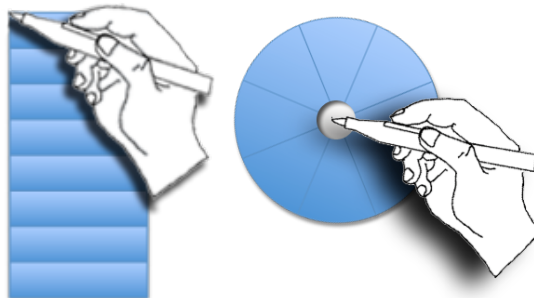


Figure 16 : Occultation de la main sur le menu. A gauche, un menu linéaire, à droite, un Marking menu.

2.1.4 Imprécision de l'interaction au doigt

Certains dispositifs permettent à l'utilisateur d'interagir directement avec le doigt (index ou pouce) comme les tableaux et tables interactives ou la plupart des SmartPhones (iPhone). En plus des précédentes difficultés (absence de clic droit, absence de raccourcis claviers, occultation), l'interaction au doigt est peu précise [Pahri 06, Roudaut 08] : le doigt a une surface de contact plus large qu'un stylet et avec certains systèmes d'exploitation comme Windows

Mobile, souvent plus large que les items eux mêmes. Ceci rend difficile la sélection précise d'un item des menus linéaires sur ces plateformes.

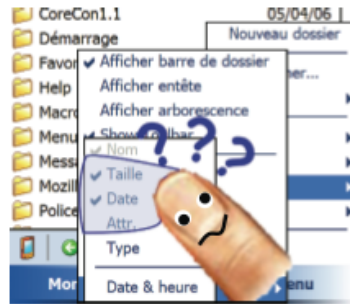


Figure 17 : Imprécision du pouce pour sélectionner précisément un item dans un menu sur dispositif mobile. Illustration extraite de [Huot 07].

Une solution triviale consiste à agrandir la taille des items mais engendre une forte sous-utilisation de l'écran : peu d'items sont affichés par rapport à ce que l'utilisateur pourrait normalement voir. D'autres solutions plus efficaces ont été proposées comme la projection de curseur (« offset cursor » [Potter 88]), Shift [Vogel 07], Thumbspace [Karlson 07], MagStick [Roudaut 08], TapTap [Roudaut 08] et escape [Yatani 08] dont nous pouvons trouver une revue dans [Roudaut 08]. Parmi ces solutions, seul le mécanisme de *projection du curseur* (« offset cursor » [Potter 88]) a été utilisé pour les menus. Ce mécanisme déplace le curseur afin que celui-ci ne se retrouve pas masqué par le doigt de l'utilisateur. Il stabilise également le curseur afin d'éliminer les problèmes d'imprécision. Cette solution a été adoptée par l'*ArchMenu* et le *ThumbMenu* [Huot 07], illustrés Figure 18. Cependant la projection de curseur a l'inconvénient d'être lent et de perdre le caractère direct de l'écran tactile.

Enfin, une solution consiste à utiliser un geste (interaction gestuelle) plutôt qu'un « tap » [Bailly 08b, Yatani 08] pour sélectionner un item. En dessinant une marque, le système reçoit davantage d'événements utilisateur. Cette augmentation d'informations en entrée est alors plus facile à interpréter correctement.

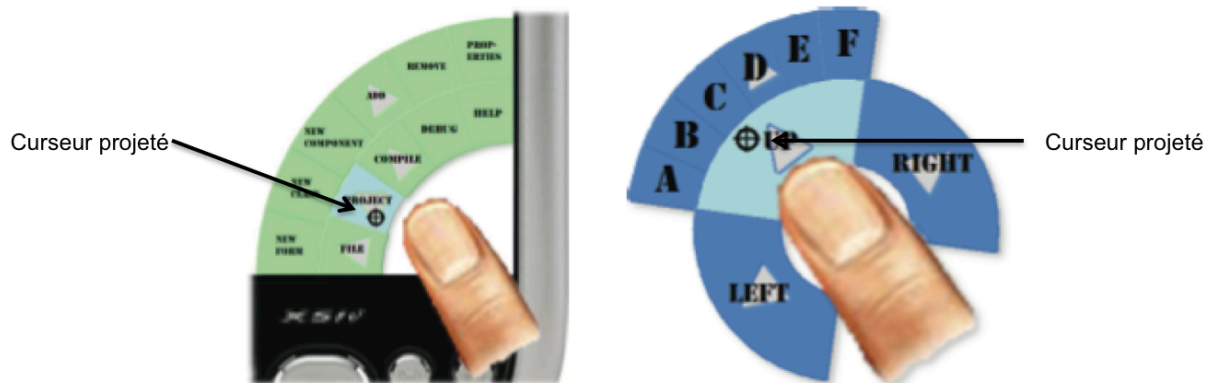


Figure 18 : A gauche, ArchMenu ; a droite, ThumbMenu [Huot 07]. Ces deux menus utilisent le mécanisme de projection du curseur (« offset cursor ») pour éviter les problèmes d'occultation et d'imprécision (illustration extraite de [Huot 07]).

Tandis que nous venons d'étudier des solutions de menus pour palier aux limitations de certains dispositifs d'entrée (absence de bouton droit, absence de claviers, occultation, imprécision), nous présentons maintenant des solutions qui exploitent les opportunités fournies par certains dispositifs pour améliorer une technique de menus.

2.1.5 Opportunités : spécificités de certains dispositifs

Les spécificités de certains dispositifs d'entrée peuvent servir à améliorer l'interaction avec les menus : par exemple, la molette des Macintosh est utilisée pour faire défiler le menu lorsque celui-ci est plus grand que l'espace disponible. Ceci évite à l'utilisateur d'avoir à déplacer le curseur jusqu'à la barre de défilement. La Hotbox [Kurtenbach 99], illustrée Figure 14, utilise la barre d'espace, très facile d'accès, pour activer le menu rapidement avec la main non dominante (l'autre servant à sélectionner les items). Comme expliqué précédemment, la pression du stylet sur la tablette est utilisée par les Donuts menus [Lian 05, Ren 08] et les PushMenus [Huot 08] pour augmenter le nombre d'items dans les Marking menus (section 1.1). Les *Tilt menus* [Tian 08], illustrés Figure 32, exploitent l'orientation (« tilt ») du stylet. Les Bimanual Marking menus [Odell 04], les Toolglass [Bier 93] ou son extension les T3 [Kurtenbach 97b], PadMouse [Balakrishnan 98] et TrackMouse [Martin 07] exploitent l'interaction bi-manuelle pour faciliter la sélection de commandes. Le Lucid Touch [Wigdor 07] permet à l'utilisateur d'interagir sur le contenu de l'écran en touchant l'arrière du dispositif.

Nous proposerons dans le chapitre 8 d'exploiter la technologie multi-points (« Multi-touch ») de certaines surfaces tactiles afin d'améliorer l'utilisabilité des techniques de menus.

2.2 Contraintes spatiales

L'espace écran nécessaire à l'affichage d'un menu peut être une contrainte majeure, particulièrement sur les dispositifs mobiles. En effet, l'interaction est significativement dégradée, voire impossible, lorsqu'une partie du menu est affichée en dehors de l'écran.

Dans le cas des menus linéaires, le menu est déplacé lorsqu'il est activé près d'un bord afin d'être affiché entièrement à l'écran : c'est le mécanisme de *débrayage*. Le menu apparaît à droite ou gauche du point d'activation en fonction de l'espace disponible. Le même mécanisme est utilisé pour les sous-menus (qui apparaissent à gauche ou à droite de l'item parent). De plus, comme expliqué dans la section 1.2, des barres de défilement sont parfois ajoutées lorsque le menu est trop long. Ainsi, il est toujours possible d'atteindre tous les items des menus linéaires.



Figure 19 : Un Compound Marking menu à 3 niveaux occupe trois fois l'espace horizontal qu'un menu linéaire.

Contrairement aux menus linéaires, les menus circulaires peuvent parfois apparaître en dehors de l'écran car ils sont normalement affichés centrés autour du point d'activation. Il y a donc toujours au moins la largeur d'un item à droite et à gauche de ce point ce qui n'est pas compatible avec une utilisation près d'un bord de l'écran. Notons néanmoins que les Multi-Stroke menus [Zhao 04] requièrent moins d'espace écran que les Compound Marking menus car ils superposent les sous-menus. L'occupation spatiale ne dépend donc pas de la profondeur du menu. Au contraire pour les Marking menus hiérarchiques classiques, les sous-menus sont affichés dans la direction de l'item parent. Par conséquent, un Compound Marking à 3 niveaux requiert 3 fois plus d'espace horizontal qu'un menu linéaire (Figure 19). Ce sont donc de mauvais candidats pour les petits écrans ou pour être utilisés comme menus contextuels (Figure 20).

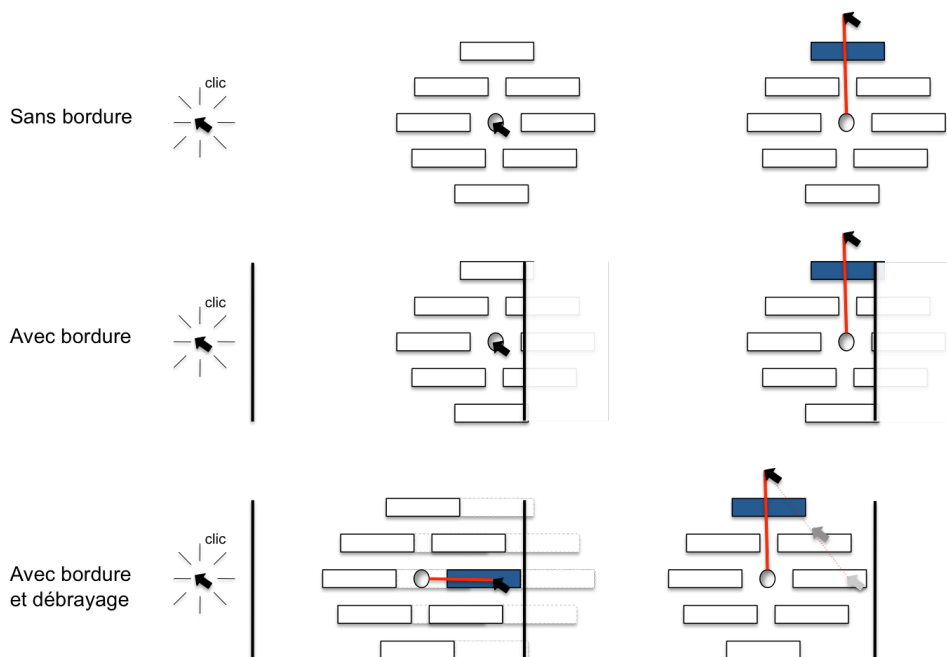


Figure 20 : En absence de bord, les Marking menus sont affichés centrés autour du point d'activation. En présence de bord, le menu risque de déborder de l'écran. Une solution est le débrayage, mais il a l'inconvénient de rompre l'association commande-geste.

Une solution, utilisée dans le logiciel Maya [Maya], consiste à utiliser un mécanisme de débrayage (Figure 20) comparable à celui des menus linéaires : Le menu est déplacé afin d'apparaître entièrement à l'écran. Cependant, cette solution a l'inconvénient de briser l'association commande-geste car le menu n'apparaît plus centré autour du point d'activation. Comme le montre la Figure 20, pour accéder à un item orienté au nord, près du bord droit de l'écran, l'utilisateur est forcé de faire un geste dans la direction Nord-Ouest.



Nous retenons pour le critère *adéquation à la plateforme* que les Marking menus sont très sensibles aux contraintes spatiales. Le chapitre 8 approfondit ce critère et décrit deux nouvelles techniques de menus :

- le leaf menu spécifiquement conçu pour dépasser les limitations des dispositifs mobiles ;
- le Multi-Touch Menu qui exploite la technologie multi-points pour la sélection de commandes avec les 5 doigts de la main.

3 Adéquation à la tâche

		Aspects	Solutions
Facteur d'applicabilité	Critère d'adéquation à la tâche	Contexte visuel	Sur demande Transient Affichage seulement des libellés Superposition Transparence Mode expert Retour visuel
		Transitions	Sur place Faible déplacement du curseur Persistance Modalité Crossing Fusion de la sélection et du contrôle continu

Tableau 4 : Tableau récapitulatif des principales propriétés et stratégies impliquées pour le critère adéquation à la tâche.

Les menus ne sont jamais l'objet principal de la tâche, mais un instrument qui aide l'utilisateur à accomplir son but (section 1 du chapitre 2). La sélection d'une commande dans un menu s'inscrit généralement dans une tâche de plus haut niveau impliquant un ou plusieurs objets d'intérêt. De ce constat, nous concluons d'une part que les menus doivent masquer le moins possible le centre d'intérêt de l'utilisateur. Nous étudions cet aspect noté "contexte visuel" dans la section 3.1. D'autre part, il est nécessaire de considérer les différentes transitions entre une technique de menus et les objets d'intérêts. CIS et SimCIS [Appert 04], présentés dans la section 1.2.1 du chapitre 3, montrent que les performances d'une technique d'interaction dépendent de la séquence d'interaction [Dillon 90, Mackay 02, Appert 04]. Les aspects de transition entre la technique de menus et son contexte d'interaction sont étudiés dans la section 3.2.

3.1 Contexte visuel

Les menus doivent masquer le moins possible la zone d'attention de l'utilisateur pour qu'il puisse se concentrer sur sa tâche principale. C'est pour cette raison que les menus, sont affichés sur « demande » [Hinckley 99] et qu'ils se ferment automatiquement après l'activation d'une commande (transience). Ceci les différencie des palettes qui masquent en permanence une partie de l'écran.

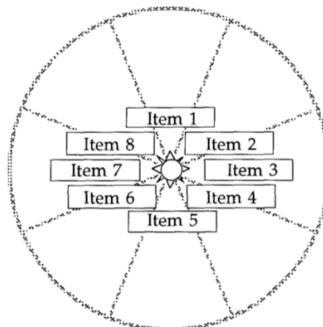
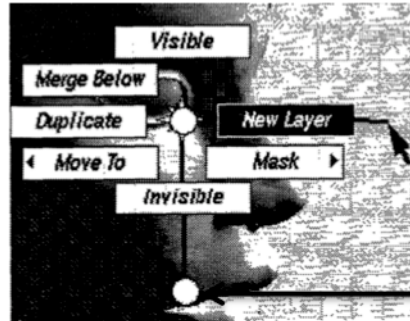


Figure 21 : En affichant seulement les labels, les Marking menus masquent moins le contexte visuel de l'utilisateur. Illustration extraite de [Tapia 95].

Pour minimiser l'espace écran occupé, une première solution consiste à afficher uniquement les libellés des commandes (« labels only ») [Tapia 95] au lieu des secteurs des Marking menus (Figure 21). Un inconvénient de cette solution est que les libellés entament sur les secteurs voisins : par exemple à la Figure 21, la partie droite de l'item 1 appartient au secteur de l'item 2. Ainsi, l'item sous le curseur peut être différent de l'item qui sera sélectionné.



Centre du menu parent

Figure 22 : Les items parents sont masqués dans certaines implémentations des Marking menus. Illustration extraite de [Tapia 95].

Une autre solution consiste à ne pas afficher les menus parents. Par exemple, Tapia [Tapia 95] affiche uniquement le centre des menus parents comme point de repère (Figure 22). Enfin, Zhao [Zhao 04] superpose les menus.

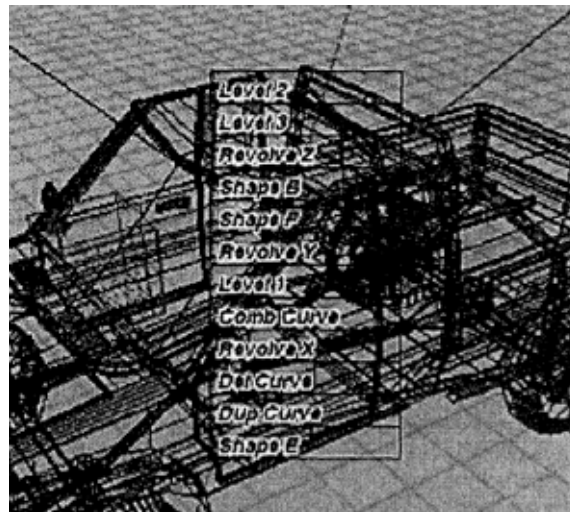


Figure 23 : Utilisation de la transparence dans le menu linéaire. La police utilisée ici est appelée « l'anti-interférence ». Illustration extraite de [Harrison 96].

La transparence constitue une autre solution qui peut aider les utilisateurs à partager leur attention entre le menu et la zone de travail [Tapia 95, Harrison 96, Hinckley 99, Rubio 02, Bowes 03]. Cependant, la transparence peut aussi entraîner des difficultés à lire le nom des items à cause des interférences avec le contexte. Harrison, [Harrison 96] propose une police, appelée « anti-interférence » qui permet d'augmenter la transparence des items sans trop dégrader leur lisibilité (Figure 23) en dessinant le contour de chaque lettre.

Finalement, le mode expert des techniques de menus offre la possibilité de ne pas afficher le menu. Les accélérateurs claviers ou les marques permettent d'activer des commandes sans masquer la zone d'attention de l'utilisateur. Il est à noter néanmoins que les raccourcis claviers peuvent conduire les utilisateurs à changer leur focus d'attention en fixant le clavier. Au contraire, le mode expert des Marking menus peut être exécuté en gardant le regard sur les objets d'intérêt. Les Marking menus, grâce à des gestes basés sur l'orientation, permettent « la sélection sans regarder » [Kurtenbach 91, Oakley 07]. Cependant, un inconvénient éventuel du mode expert est le manque de retour visuel après l'activation d'une commande : les utilisateurs ne savent pas si le système a reconnu la commande désirée lorsqu'il n'y a pas de retour visuel

sur les objets d'intérêt [Tapia 95]. Pour cela, MacOSX met en surbrillance l'item de la barre de menus qui correspond à la branche contenant la commande invoquée. Pour les Marking menus, Tapia [Tapia 95] affiche pendant quelques dixièmes de seconde la marque parfaite après l'activation de la commande.

3.2 Transitions

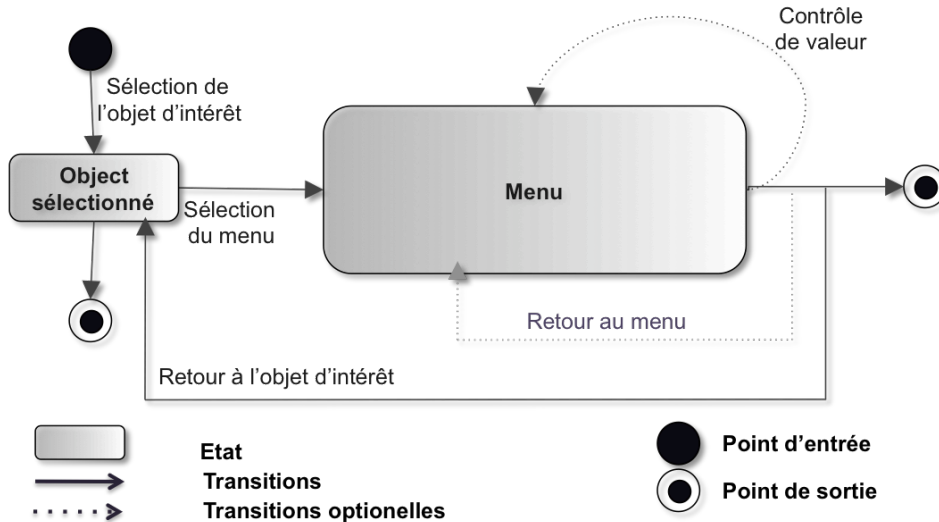


Figure 24 : les différentes transitions impliquées dans le processus de sélection d'une commande entre le menu et le contexte.

Pour accomplir une tâche, l'utilisateur est généralement amené à choisir un (ou des) objet(s) d'intérêt, à accéder au menu, à sélectionner plusieurs commandes et à contrôler leurs paramètres. La « séquence d'interaction » [Appert 04], permettant à l'utilisateur d'accomplir sa tâche, peut modifier significativement les performances d'une technique. Dillon affirme que le « vrai temps de sélection⁴ » doit prendre en compte le temps pour accéder au menu (*sélection d'un menu*) et le temps pour *revenir aux objets d'intérêt* [Dillon 90]. Nous explicitons deux autres transitions qui interviennent pour la sélection de commandes : le *retour au système de menu* et le *contrôle de valeur*. L'automate de la Figure 24 met en évidence ces quatre transitions :

- la sélection du menu (transition objet-commande) ;
- le retour à l'objet d'intérêt (transition commande-objet) ;
- le retour au menu (transition commande-commande) ;
- le contrôle de valeur (transition commande-valeur).

Nous étudions les solutions qui permettent d'améliorer chacune de ces transitions.

3.2.1 Sélection du menu (transition objet-commande)

La transition objet-commande [Cance 06] est la distance en temps et en espace pour accéder au menu à partir d'un objet d'intérêt (document, image, forme 3D, etc). Cette distance est similaire au degré d'indirection dans le modèle de l'interaction instrumentale [Beaudouin-Lafon 00] car un menu est un instrument (section 1 du chapitre 2). Elle consiste généralement à pointer un objet : objet d'intérêt si le menu est contextuel ou alors un interacteur à l'activation du menu comme une barre de menus. Les menus contextuels réduisent cette transition car ils ont la propriété d'être « sur place » [Hinckley 99]. Ils sont en effet activés et affichés dans la zone d'attention de l'utilisateur et ne nécessitent pas de déplacement particulier. Au contraire, les

⁴ The true cost of command selection.

menus attachés à la barre de menus (ou d'autres systèmes de commandes comme les palettes) sont plus éloignés de l'objet d'intérêt et plus longs à accéder.

3.2.2 Retour à l'objet d'intérêt (transition commande-objet)

Les utilisateurs sont souvent amenés à revenir à l'objet d'intérêt après avoir activé une commande : c'est la transition commande-objet [Cance 06]. Cette transition peut être particulièrement importante (en temps ou en espace) pour certains menus. Par exemple, un Compound Marking menu [Kurtenbach 93a] avec 3 niveaux de profondeur (Figure 25) force l'utilisateur à parcourir une distance deux fois plus grande que les menus linéaires pour revenir au point d'activation du menu donc à l'objet d'intérêt. Deux stratégies ont été utilisées pour minimiser cette transition.

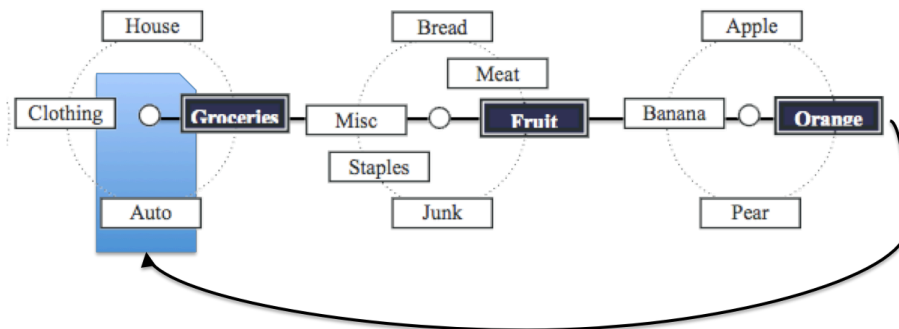


Figure 25 : Transition commande-objet avec les Compound Marking menus [Kurtenbach 93a]. Après avoir sélectionné Orange, l'utilisateur doit parcourir une distance importante pour revenir à l'objet d'intérêt (schématisé par le rectangle bisoté).

La première consiste à déplacer le menu au lieu de déplacer le curseur (Figure 26). Basé sur ce principe, les Slippery menus [Cance 06] glissent en dessous du curseur qui reste fixe. A la fin de l'interaction, celui-ci est toujours au point d'activation, supprimant ainsi la transition commande-objet.

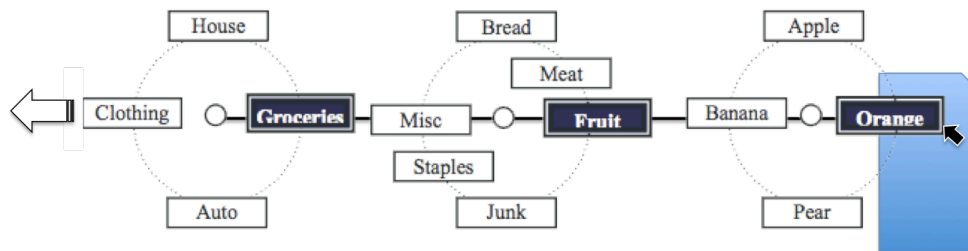


Figure 26 : Transition commande-objet avec les Slippery menus [Cance 06]. Ce menu glisse sous le curseur afin que celui-ci reste à la position de départ (au dessus de l'objet d'intérêt).

La seconde alternative consiste à superposer les sous-menus. Ainsi, le curseur ne s'éloigne jamais très loin de l'objet d'intérêt comme avec les Multi-Stroke menus [Zhao 04], illustrés Figure 27.

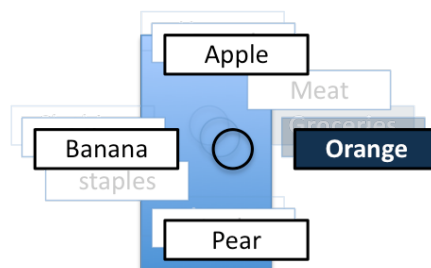


Figure 27 : Transition commande-objet avec les Multi-Stroke menus [Zhao 04]. Les sous-menus se superposent afin que le curseur ne s'éloigne jamais de l'objet d'intérêt.

3.2.3 Retour au menu (transition commande-commande)

L'utilisateur souhaite parfois appliquer plusieurs commandes différentes à un même objet ou appliquer une même commande à plusieurs objets différents (respectivement développement et factorisation dans CIS [Appert 04]). Il est alors forcé de faire plusieurs transitions objet-commande et commande-objet.

De ce point de vue, les palettes à état offrent un avantage par rapport aux menus car elles permettent de retenir le mode (conservation de l'outil) après l'activation d'une commande. L'utilisateur peut alors appliquer plusieurs fois la même commande à plusieurs objets sans la ré-sélectionner grâce à la propriété de persistance. Cependant, Raskin [Raskin 00] souligne que la persistance du mode peut entraîner des erreurs lorsque l'utilisateur oublie quel est le mode courant.

Certains menus ont été modifiés afin de faciliter la sélection de plusieurs commandes sur un même objet en laissant le menu ouvert après l'activation d'une commande⁵. Par exemple, les Tear-Off menus [Lee 93] peuvent être détachés du menu parent afin d'éviter à l'utilisateur de retraverser la hiérarchie de commandes. Les *Floating Pie menus* [Rubio 02], illustrés Figure 28, reposent sur la même idée, un bouton situé au centre rendant un sous-menu persistant.

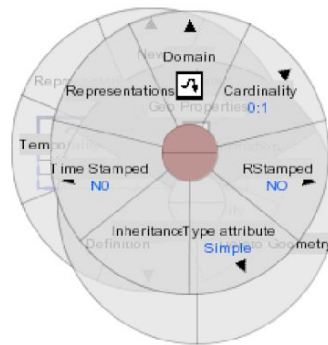


Figure 28 : Les Floating Pie menus [Rubio 02]. L'utilisateur peut empêcher la fermeture d'un sous-menu grâce à un bouton de « lock » au centre du menu. Illustration extraite de [Rubio 02].

Finalement, l'activation par « crossing », qui active un élément en le traversant de part en part (section 1.4 du chapitre 4), a été utilisée pour activer plusieurs commandes dans les menus linéaires en un seul geste. Dans *CrossY* [Apitz 04], illustré Figure 29, l'utilisateur peut choisir plusieurs paramètres (correspondant généralement à plusieurs commandes) à son outil de dessin en un seul geste. Cette propriété autorise une « composition fluide de commandes » en exécutant un geste unique.

⁵ Nous notons que dans ce cas, cet interacteur ne vérifie plus totalement notre définition du menu (section 1 du chapitre 2) car il n'est plus transient. Dans la mesure où ce n'est pas son mode de fonctionnement par défaut, mais que l'utilisateur choisit explicitement quand le menu devient persistant, nous les intégrons dans notre espace MenUA.

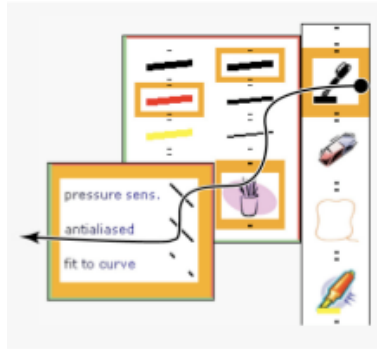


Figure 29 : Dans CrossY [Apitz 04], les utilisateurs peuvent sélectionner plusieurs commandes en un seul geste grâce à la « composition fluide de commandes ». Illustration extraite de [Apitz 04].

3.2.4 Contrôle de valeurs (transition commande-valeur)

Le dernier type de transitions est le contrôle de valeurs. Dans certains cas, il est utile de contrôler une valeur après l'activation d'une commande. Par exemple, une fois la commande « Zoom » activée, l'utilisateur peut souhaiter indiquer la valeur du zoom (augmenter ou diminuer le zoom). Lorsque ces deux opérations peuvent être effectuées dans le même geste, c'est la « fusion de la sélection de commandes et de la manipulation directe » [Guimbretiere 05]. Plusieurs techniques de menus ont cette caractéristique.

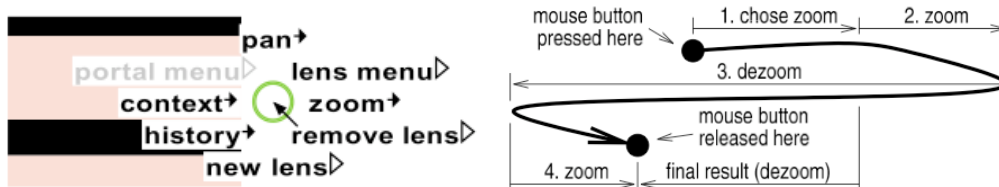


Figure 30 : Control menus [Pook 00]. L'utilisateur sélectionne la commande zoom en faisant un geste en direction de l'item. Un fois une distance seuil dépassée, la commande est activée et l'utilisateur contrôle continûment le niveau de zoom jusqu'à ce qu'il relâche le bouton de la souris. Illustration extraite de [Pook 00].

Les Control menus [Pook 00], illustrés Figure 30, sont des menus circulaires basés sur le « crossing ». Lorsque l'utilisateur a dépassé une distance seuil, la commande est activée, et le contrôle de l'action commence. Les Flowmenus [Guimbretiere 00], illustrés Figure 31, nécessitent de quitter la zone centrale du menu pour sélectionner une commande et d'y revenir pour l'activer et commencer l'opération. Le fait que l'opération ne commence pas au point d'activation du menu est parfois un inconvénient comme pour le dessin d'une forme libre. Au contraire, les Tilt menus [Tian 08] permettent de sélectionner une commande sans quitter le point d'activation. En effet, c'est l'angle du stylo sur la table détermine la commande sélectionnée (Figure 32). Le contrôle de valeur est alors exécuté en déplaçant la souris.

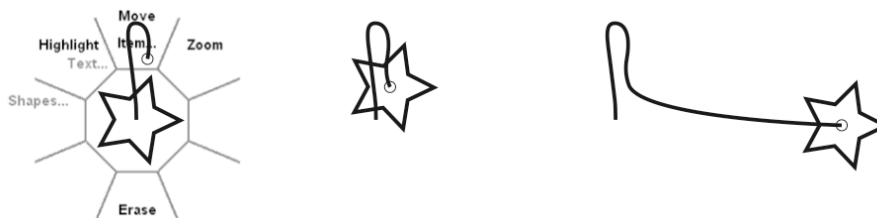


Figure 31 : Flowmenus [Guimbretiere 00]. Après avoir sélectionné la commande « Move », l'utilisateur revient dans la zone centrale pour déplacer l'objet d'intérêt. Illustration extraite de [Guimbretiere 00]

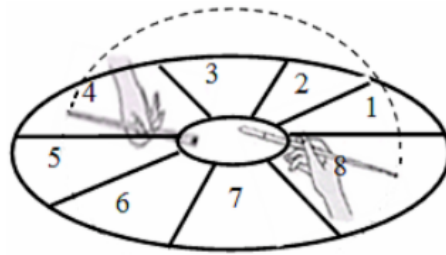


Figure 32 : Tilt menus [Tian 08]. L'utilisateur sélectionne une commande en inclinant le stylet dans une direction. Une fois la commande sélectionnée, il peut commencer l'opération en déplaçant le stylet à partir du point d'activation. Illustration extraite de [Tian 08].

Enfin, les Toolglass [Bier 93], les Tracking menus [Fitzmaurice 03] et les Pie cursor [Fitzmaurice 08], bien que pas considérés comme des menus dans ce manuscrit, ont la particularité de sélectionner un objet d'intérêt et une commande en une seule opération grâce au paradigme du « clic à travers ». Pour cela, l'utilisateur positionne l'item correspondant à la commande, qui est semi-transparent, au dessus de l'objet d'intérêt. Un clic à cette position sélectionne alors à la fois l'item et l'objet qui est en dessous. Ainsi, l'utilisateur peut commencer le contrôle d'une valeur à l'endroit précis où il le souhaite.

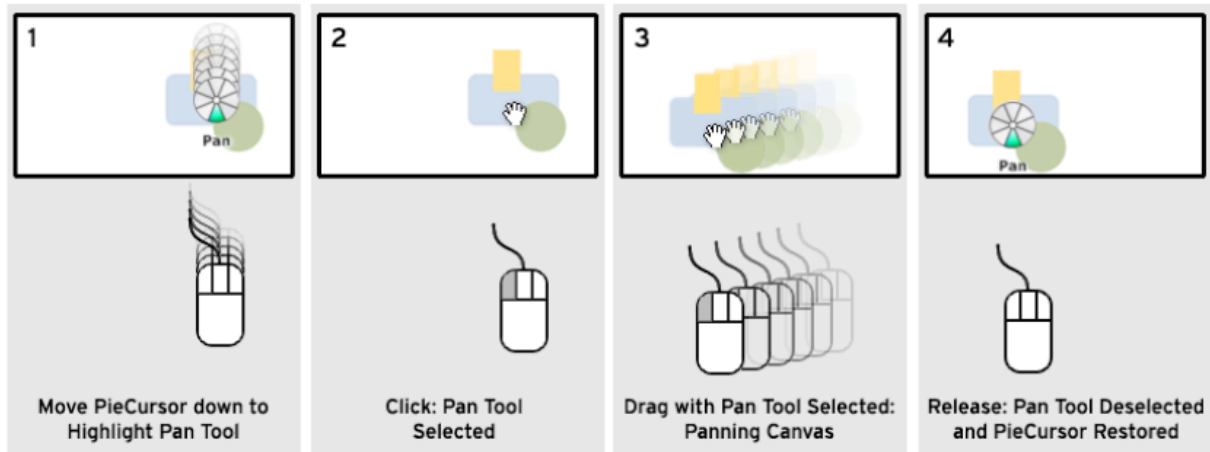


Figure 33 : PieCursor [Fitzmaurice 08]. 1) l'utilisateur positionne la commande « Pan » au dessus du groupe d'objets. 2) Il sélectionne à la fois le groupe d'objets et la commande Pan avec un seul « press », 3) Il contrôle la valeur du pan en déplaçant les objets. 4) Enfin, il relâche le bouton de la souris pour terminer l'opération. Illustration extraite de [Fitzmaurice 08].

La Figure 33 illustre la sélection de la commande « Pan » et du contrôle de sa valeur avec un Tracking menu. L'utilisateur positionne d'abord le secteur associé à la commande « Pan » au dessus du groupe d'objets. Un appui sur le bouton de la souris sur ce secteur sélectionne le groupe d'objets et active le mode « Pan ». Un déplacement de la souris sur le drag entraîne alors un déplacement du groupe d'objets.



Nous retenons que l'occupation spatiale des Marking menus hiérarchiques implique :

- l'exploitation de traits visuels comme la transparence ou l'affichage des items uniquement, pour ne pas perdre le contexte de l'objet d'intérêt ;
- l'étude des transitions entre le menu et l'objet d'intérêt.

4 Conclusion

Dans ce chapitre, nous avons étudié les techniques de menus vis-à-vis du facteur d'applicabilité, la seconde facette de notre espace de critères MenUA.

Les trois critères d'adéquation, *application*, *plateforme* et *tâche*, fournissent un point de vue complémentaire aux critères d'utilisabilité du chapitre 4. Par exemple, nous avons montré de nombreux avantages liés aux Marking menus pour les critères d'utilisabilité (chapitre 4). Au contraire, le facteur d'applicabilité a mis en évidence les limites de cette technique de menus vis-à-vis de certains critères comme l'*adéquation à l'application* et son nombre de commandes.

Cette étude nous a permis d'illustrer tous les aspects proposés par les menus de la littérature. Nous présentons en annexe l'inverse de cette analyse. Chacun des menus est décrit selon les différents aspects définis dans MenUA. Cette étude complémentaire permet aux concepteurs d'applications de comparer plus facilement plusieurs techniques.

Une perspective à ce chapitre consisterait à introduire un 4^{ème} aspect orthogonal aux 3 premiers, l'*adéquation à l'utilisateur*, qui prendrait en compte l'adéquation d'une technique de menus vis-à-vis de certaines catégories d'utilisateurs comme les personnes âgées, les enfants ou les personnes handicapées. Par exemple, les menus vocaux/audios [Zhao 07, Friedlander 98] peuvent être plus adaptés que les menus basés sur les dispositifs traditionnels (souris, claviers, écrans) pour des personnes atteintes de déficiences visuelles. De même, les menus avec des libellés iconiques, plutôt que textuels, seraient préférables pour les enfants. L'accessibilité est un aspect rarement considéré dans la littérature sur les menus qui méritait davantage d'attention.

Dans la seconde partie de ce mémoire, nous revenons sur les Marking menus vis-à-vis du critère d'*adéquation à l'application*. Nous proposons dans le chapitre 6 une solution pour le problème de la profondeur et dans le chapitre 7 une solution pour celui de la largeur.

Synthèse de la partie I

La première partie de ce manuscrit a été consacrée à la définition, la classification et la caractérisation des techniques de menus existantes.

Après avoir précisé la terminologie adoptée et introduit les concepts utiles à l'étude des menus (**chapitre 2**), nous avons introduit notre espace de conception appelé MenUA que nous avons positionné par rapport aux espaces de conception existants (**Chapitre 3**). Notre espace MenUA est une taxonomie des travaux sur les menus qui repose sur une liste de *critères* organisés en deux *facteurs* complémentaires : l'utilisabilité (**chapitre 4**) et l'applicabilité (**chapitre 5**). Pour chacun des critères, un ensemble de *solutions* ont été explicitées et illustrées par des techniques de menus de la littérature.

Notre espace MenUA nous a permis de dresser un état de l'art des travaux sur les menus vis-à-vis de nos critères. Ces critères ont également permis de souligner des compromis à faire en conception, comme le balancement entre minimiser l'espace occupé par le menu (critère *adéquation à la plateforme*) vs. favoriser la navigation dans le menu (critère *rapidité et précision*), qui sont utiles aux concepteurs d'applications pour faire des choix informés. Enfin, MenUA nous a permis de mettre en évidence des éléments moins étudiés qui ont motivé la conception de 4 nouvelles techniques de menus dans la seconde partie de ce manuscrit.

L'étude du critère *rapidité et précision* a montré que peu de travaux se sont intéressés aux performances des utilisateurs utilisant des chemins détournés pour accéder à une commande. Aussi, le **chapitre 6** est consacré à l'exploration dans les systèmes de menus. Nous présentons puis évaluons les Wave menus, une nouvelle technique de menus qui améliore le mode novice des Marking menus.

L'étude du critère *adéquation à l'application* a mis en évidence les limites des Marking menus pour contenir un nombre important de commandes en largeur. Des solutions ont été proposées, mais elles ne prennent pas explicitement en considération le critère *apprentissage et mémorisation*, un critère encore assez peu étudié. Nous présentons au **chapitre 7** les Flower menus, une nouvelle technique de menus qui augmente le nombre de commandes possibles des Marking menus sans en dégrader l'apprentissage du mode expert (critère *apprentissage et mémorisation*).

L'étude du critère *adéquation à la plateforme* a mis en évidence la difficulté d'utilisation de certaines techniques avec certains dispositifs. Elle a également souligné l'intérêt d'exploiter des dispositifs particuliers pour améliorer l'utilisabilité du menu. Aussi, nous présentons deux nouvelles techniques de menus dans le **chapitre 8** : le Leaf menu qui augmente les menus linéaires de raccourcis gestuels pour faciliter la sélection de commandes sur dispositifs mobiles et le Multi-Touch Menu (MTM) qui exploite la technologie multi-points pour favoriser la navigation et la sélection de commandes.

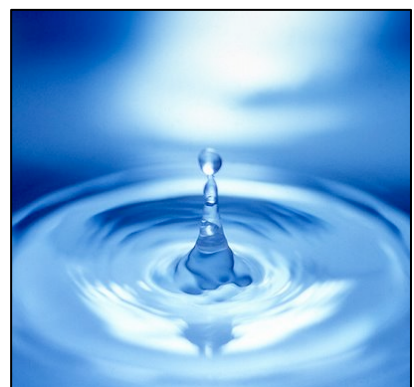
Ces quatre nouvelles techniques de menus (Wave menu, Flower menu, Leaf menu et MTM) sont présentées dans la suite de ce manuscrit vis-à-vis de notre espace MenUA.

Conception et Evaluation des Techniques de Menus

Partie II

Exploration dans les menus hiérarchiques : Les Wave menus, une solution pour les Marking menus

Chapitre 6



1	EXPLORATION DANS LES MENUS	114
1.1	EXPLORATION DANS LES SYSTEMES D'INFORMATION	114
1.2	TACHE DE FEUILLETAGE	116
1.3	TACHE DE RECHERCHE	116
1.4	RECHERCHE VISUELLE	119
1.5	SOLUTIONS CONCEPTUELLES POUR L'EXPLORATION	122
2	WAVE MENUS	126
2.1	RAPPEL DES LIMITATIONS DES MARKING MENUS HIERARCHIQUES	126
2.2	PRINCIPE DES WAVE MENUS	127
2.3	PROPRIETES DES WAVE MENUS	129
2.3.1	RAPIDITE ET PRECISION	129
2.3.2	APPRENTISSAGE ET MEMORISATION	132
2.3.3	SATISFACTION	132
2.3.4	ADEQUATION A L'APPLICATION	133
2.3.5	ADEQUATION A LA PLATEFORME	133
2.3.6	ADEQUATION A LA TACHE	134
2.4	WAVE MENUS INVERSEES	136
3	EVALUATION	137
3.1	DIFFICULTES DE L'EVALUATION POUR LES CHEMINS DETOURNES	137
3.2	DEFINITION ET REALISATION D'UNE TACHE DE NAVIGATION	138
3.3	EXPERIENCE	140
3.4	RESULTATS	142
3.5	DISCUSSIONS	144
4	CONCLUSION	146

L'exploration est une tâche fréquente dans les systèmes d'information. Les utilisateurs sont souvent amenés à naviguer, feuilleter ou rechercher de l'information dans des structures complexes tels que les réseaux sociaux (www.facebook.fr), les sites marchands, les moteurs de recherche (www.google.com), les graphes scientifiques (arbres des espèces) ou les systèmes de menus. Nous imaginons aisément que les utilisateurs adoptent des stratégies de recherche différentes selon le système d'information considéré.

Contrairement aux systèmes de visualisation d'arbres (Treemap [Johnson 91], Hyperbolic view, [Lamping 95], SpaceTree [Plaisant 02], etc), les menus ne sont pas les principaux objets d'intérêt de l'utilisateur. De plus, les menus manipulent des hiérarchies relativement petites (quelques centaines de nœuds). Ces hiérarchies sont souvent délicates à concevoir [Snyder 85] car les utilisateurs finaux peuvent être variés (section 3 du chapitre 2). Si elle ne correspond pas à leur modèle mental, les utilisateurs risquent d'emprunter des *chemins détournés* (section 4 du chapitre 2). Ce type de chemin a reçu moins d'attention que les autres (droits ou directs) dans la littérature sur les menus (section 4 du chapitre 2). Pourtant, il est utile de rappeler que de nombreux utilisateurs naviguent dans une partie du menu avant d'atteindre la commande désirée. Même les utilisateurs expérimentés, qui connaissent le chemin pour accéder à quelques commandes fréquentes (moins de 10% d'après [Linton 99]), peuvent emprunter des chemins détournés pour les commandes moins fréquentes. Le mode novice, permettant de naviguer dans les systèmes de menus, est donc certainement plus utilisé que le mode expert. Pourtant, la plupart des travaux sur les Marking menus portent sur le mode expert. En définitive, les gains obtenus sont assez faibles (en dixième de secondes) et ne concernent qu'une minorité d'utilisateurs expérimentés. Au contraire, une amélioration du mode novice profite à un public beaucoup plus vaste et le gain en temps peut être important (plusieurs secondes comme nous le verrons dans la section 3). L'importance du mode novice et la prise en compte des chemins détournés ne doivent donc pas être sous-estimées. Une amélioration minime de la navigation dans les menus peut augmenter la productivité de manière significative. C'est pourquoi, nous proposons dans ce chapitre de détailler l'exploration dans les systèmes de menus que nous avons déjà caractérisé par des aspects dans MenUA au chapitre 4, section 1.2 et 1.3.

La section 1 est consacrée à la notion d'exploration dans les systèmes d'information. Nous exposons un modèle d'exploration de la littérature permettant de fixer la terminologie utilisée. Nous identifions alors deux tâches de navigation associées aux menus : le *feuilletage* et la *recherche*. Ces deux tâches nous permettent de mieux cerner la manière dont les utilisateurs naviguent dans les menus ainsi que leurs besoins interactionnels. Cette analyse met en évidence 3 propriétés du mode novice qui favorisent la navigation dans une hiérarchie de commandes.

La section 2 est dédiée au cas particulier des Marking menus. Ces menus souffrent de plusieurs limitations lorsqu'ils sont hiérarchiques (chapitre 4). Alors qu'une technique a été proposée pour améliorer le mode expert, aucune solution n'a été apportée pour le mode novice. Nous proposons les « Wave menus », une nouvelle technique qui améliore l'exploration dans les Marking menus. Les Wave menus sont présentés au regard de notre espace de conception MenUA.

La section 3 traite du problème de l'évaluation d'une technique pour une tâche de navigation. Nous explicitons les difficultés et proposons une tâche compatible avec une expérimentation contrôlée afin de comparer les performances de trois techniques de menus : Compound, Multi-Stroke et Wave menus.

1 Exploration dans les menus

Bien que l'exploration dans les documents hypermédias soit étudiée depuis longtemps [Waterworth 91, Norman 91, Nigay 98, Card 99], l'exploration dans les menus l'est beaucoup moins. Afin de mieux comprendre cette notion, nous exposons deux modèles d'exploration dans les systèmes d'information que nous transposons ensuite aux menus.

1.1 Exploration dans les systèmes d'information

Le terme « Exploration » n'a pas, à notre connaissance, de définition précisément établie [Waterworth 91, Carmel 92, Nigay 98]. Un ensemble de verbes réfèrent à cette notion (explorer, déambuler, naviguer, rechercher, feuilleter, faire une requête, parcourir, surfer) dont certains sont utilisés indifféremment dans la littérature [Waterworth 91]. Le modèle de Waterworth [Waterworth 91] et le modèle de Nigay [Nigay 98] nous aident à fixer la terminologie.

Modèle d'exploration de Waterworth [Waterworth 91]

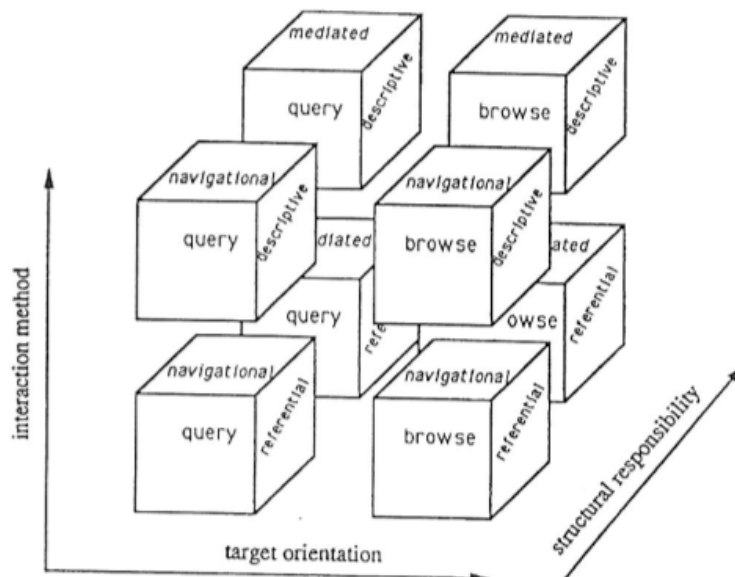


Figure 1 : Modèle d'exploration des systèmes d'information. Illustration extraite de [Waterworth 91].

Waterworth [Waterworth 91] propose un modèle d'exploration selon trois axes (Figure 1) :

- responsabilité (« structural responsibility ») ;
- précision du but (« target orientation ») ;
- méthode interactionnelle (« interaction method »).

Responsabilité : cet axe permet de définir si la recherche d'information est dirigée par le système ou par l'utilisateur. Il différencie le terme *Interrogation* (« mediated ») du terme *Navigation* (« navigational »)¹. Dans un moteur de recherche, qui considère en entrée une requête, c'est le système qui effectue l'opération de recherche et de présentation du résultat à l'utilisateur. Au contraire, dans les systèmes de menus, l'utilisateur est responsable du processus de recherche et se déplace de menus en menus. L'exploration des menus est donc limitée au cas de la navigation.

¹ Les termes utilisés dans la littérature pour différencier si la responsabilité est du côté du système ou de l'utilisateur sont : mediated vs navigation dans [Waterworth 91] ou querying vs navigation dans [Abowd 92] et [Furnas 97]

Précision du but : nous avons déjà présenté cet axe dans la section 3 du chapitre 2 pour les systèmes de menus. Nous rappelons que cet axe définit si l'utilisateur a un but précis. Waterworth distingue uniquement deux cas : le *Feuilletage* (« browsing ») et la *Recherche* (« querying »). Le feuilletage se définit par l'absence d'une cible de recherche². En s'appuyant sur les travaux de Norman [Norman 91], nous avons affiné le cas « recherche » en : *recherche d'une fonctionnalité* et *recherche d'une commande*. Dans le second cas, l'utilisateur connaît le nom exact de l'item recherché.

Méthode interactionnelle : Cet axe définit la manière dont l'interface va permettre à l'utilisateur de rechercher l'information. Il oppose les termes *Description* (« descriptive ») où l'utilisateur doit décrire ce qu'il recherche (comme dans les moteurs de recherche) de l'*Extraction* (« retrieval ») où l'utilisateur sélectionne ce qu'il veut comme dans les systèmes de menus. L'exploration des menus est donc limitée au cas extraction.

Comme dans [Nigay 98], pour le cas de la visualisation de grands espaces informationnels, nous retenons de ce modèle deux tâches de navigation pour les systèmes de menus : feuilletage et recherche. A la section 3, du chapitre 2, nous avons affiné la tâche de recherche en : *recherche d'une fonctionnalité* et *recherche d'une commande* (Figure 2).

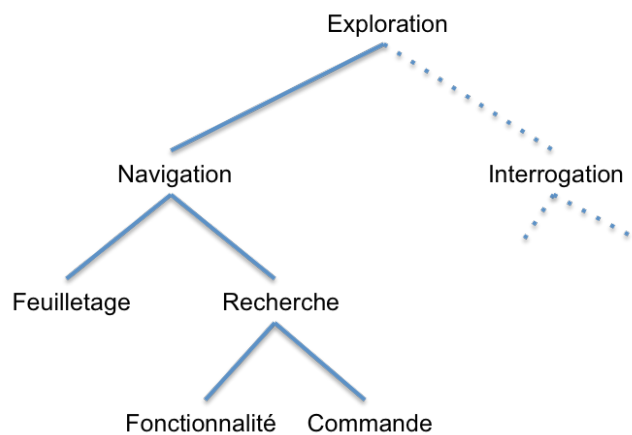


Figure 2 : Structuration de l'exploration pour les systèmes de menus.

D'autres types de tâches ont été répertoriés dans le cas de la visualisation d'information [Plaisant 02, Lee 06] comme les tâches de topologie (par exemple, déterminer la profondeur d'un arbre, le nombre de nœuds voisins, le chemin le plus court entre deux nœuds, etc). Nous ne retenons pas ces tâches pour le cas des menus car ces derniers ne sont pas l'objet central d'intérêt de l'utilisateur (objet de la tâche dans [Dubois 01]). Notons néanmoins que la tâche de "première visite", qui consiste à retrouver un élément pour la première fois, est souvent différenciée de la "tâche de revisite" qui consiste à retrouver cet élément. Nous considérons cette distinction en définissant deux tâches de recherche : recherche d'une fonctionnalité (première visite) et recherche d'une commande (revisite).



Nous retenons que l'exploration dans les systèmes de menus peut être décomposée en deux tâches : le *feuilletage* et la *recherche*.

1.2 Tâche de feuilletage

Dans une tâche de feuilletage, l'utilisateur ne recherche pas un élément en particulier. Il est davantage dirigé par la découverte du menu que par la sélection d'une commande. Il peut quitter le menu sans avoir activé une commande. La tâche de feuilletage est souvent associée à un utilisateur inexpérimenté. Lorsque celui-ci prend en main un nouveau logiciel, il souhaite avoir un aperçu rapide de ses possibilités, c'est-à-dire connaître les commandes à sa disposition [Norman 91]. En d'autres termes, il désire obtenir une vue globale du système de menus (principales fonctionnalités, etc).

La tâche de feuilletage se caractérise par une navigation : a) rapide, b) en largeur et c) en haut de la hiérarchie. Les utilisateurs lisent peu d'items. Ils se concentrent principalement sur les titres des sous-menus afin d'avoir une idée rapide de leur contenu. La navigation est rarement en profondeur car à ce stade, les utilisateurs sont rarement intéressés par les détails. Ils explorent donc surtout le haut de la hiérarchie en passant rapidement d'un menu à son voisin.



Nous retenons que la tâche de feuilletage se caractérise par une navigation rapide, en largeur et en haut de la hiérarchie. Elle nécessite de pouvoir passer rapidement et sans effort d'un menu à son voisin.

1.3 Tâche de recherche

La recherche d'une commande ou d'une fonctionnalité est la tâche principale dans un système de menus. Comme nous l'avons vu dans la section 1 du chapitre 4, l'organisation du menu (structure, vocabulaire, etc) affecte le nombre de branches visitées par l'utilisateur.

Afin de mieux définir la recherche d'une commande ou d'une fonctionnalité, nous introduisons la notion de *familiarité* [Norman 91]. La familiarité d'un utilisateur avec une partie (ou la totalité) d'un menu détermine la « quantité d'exploration » [Appert 07], c'est-à-dire le nombre de branches visitées. Moins l'utilisateur est familier avec un menu, plus il en explore une large partie et est amené à remonter fréquemment dans la hiérarchie. A l'opposé, un utilisateur familier tend à adopter un chemin quasi optimal (ou droit) pour accéder à une commande. La familiarité aide à prédire le temps, la difficulté et l'effort nécessaires pour trouver une commande. K. Norman définit la familiarité d'un utilisateur avec un menu comme une fonction de la précision du but (fonctionnalité ou commande) de l'utilisateur, de son expertise, et de la profondeur du menu. La fonction de familiarité s'exprime ainsi :

$$\text{Familiarité}_{(u,m)} = f(\text{Connaissance}_{(u,m)}, \text{Expertise}_{(u,m)}, \text{Profondeur}_{(m)})$$

Avec u : utilisateur et m : menu

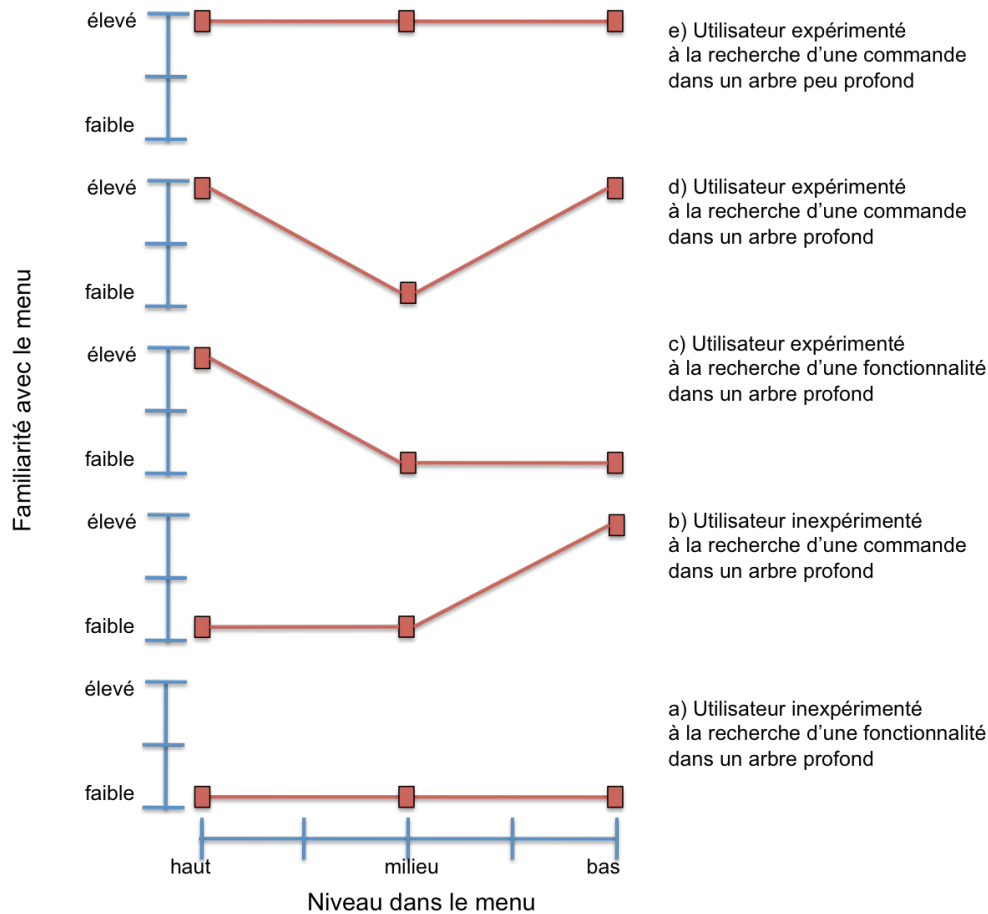


Figure 3 : Familiarité d'un utilisateur en fonction de la profondeur du menu, de son expertise et la précision de son but. Illustration réalisée à partir de [Norman 91].

La Figure 3 extraite de [Norman 91] montre la relation entre la familiarité de l'utilisateur avec des menus contenant un grand nombre de commandes en fonction du niveau de profondeur où se trouve l'utilisateur (haut, intermédiaire et bas) et de 4 couples (Expertise, précision de but)

Utilisateur inexpérimenté qui recherche une fonctionnalité

L'utilisateur est inexpérimenté et ne connaît ni le nom ni le moyen d'accéder à la fonctionnalité désirée. Le titre des sous-menus l'aide peu quelle que soit la profondeur. Il risque de fortement naviguer et le chemin sera fortement détourné. Il peut se perdre dans la structure, ne plus savoir où il se trouve et visiter plusieurs fois la même branche. Il est à noter qu'il peut passer au dessus de l'item recherché sans s'apercevoir qu'il correspond à la fonctionnalité désirée. C'est le cas a) de la Figure 3. Quelle que soit la profondeur où se trouve l'utilisateur, celui-ci n'est pas familier avec le menu.

Illustrons ce cas sur l'exemple de la Figure 4 qui montre la hiérarchie de Stata, un logiciel de statistique. Un étudiant découvre à la fois les statistiques et ce logiciel. Son objectif est en fait de sélectionner la commande « tabstat », mais il ne connaît pas ce nom. Il sait juste que c'est une opération en relation avec « la moyenne ». Il risque d'explorer une grande partie de l'arbre de commandes avant d'arriver à « tabstat » car plusieurs éléments sont en relation avec la notion de « moyenne ».

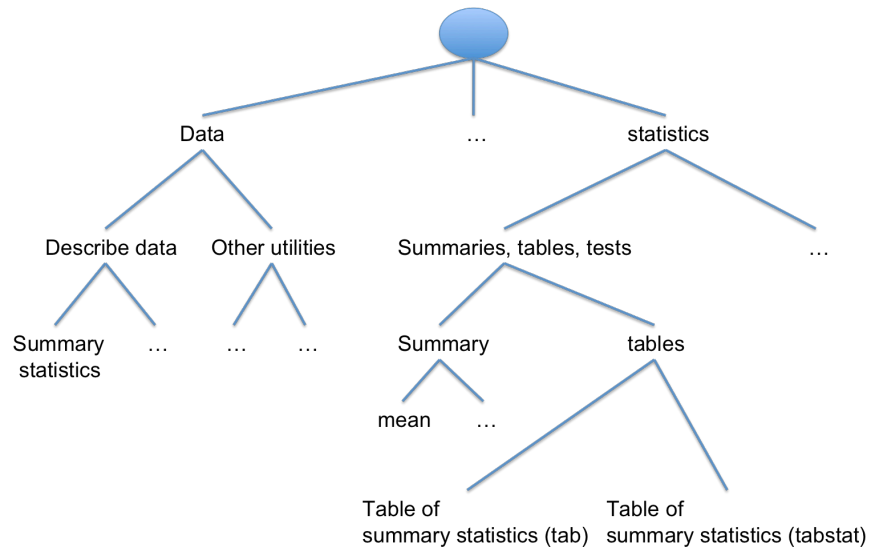


Figure 4 : Arbre de commande simplifié du logiciel de statistiques « Stata ».

Utilisateur inexpérimenté qui recherche une commande

L'utilisateur est toujours inexpérimenté, mais il connaît maintenant le nom de la commande recherchée. Il l'a certainement déjà sélectionnée préalablement, mais a oublié comment y accéder. L'utilisateur risque toujours d'emprunter un chemin détourné, mais plus il se rapproche de la commande, plus la structure du menu va le guider vers son but. En effet, plus on descend dans la hiérarchie, plus les titres des sous-menus sont précis car ils sont associés à un nombre moins important de commandes [Norman 91]. C'est le cas b) de la Figure 3. L'utilisateur est plus familier dans le bas de la hiérarchie où les titres des sous-menus sont plus concrets et donc plus faciles à comparer au nom de la commande recherchée.

L'étudiant ne connaît toujours pas « Stata », mais a retenu de son cours qu'il devait utiliser la commande « tabstat ». Il ne sait pas où celle-ci est localisée. Il risque de commencer par se tromper, mais une fois dans « statistics », il se laissera guider par les titres des sous-menus jusqu'à arriver à « tabstat ».

Utilisateur expérimenté qui recherche une fonctionnalité

Un utilisateur expérimenté connaît bien les premiers niveaux de la hiérarchie, car ce sont ceux qu'il visite le plus souvent. Ainsi, lorsqu'il recherche une nouvelle fonctionnalité, il peut éliminer des branches non-pertinentes dans la partie haute du menu (contrairement au cas a). Cependant, il risque toujours de naviguer dans les parties basses de la hiérarchie qu'il ne connaît pas. C'est le cas c) de la Figure 3 où l'utilisateur est surtout familier avec les branches près de la racine.

L'étudiant est habitué à « Stata » pour éditer ses données numériques, mais décide pour la première fois de faire « une moyenne » sur ces données. Il sait que cette fonctionnalité n'est pas dans la branche « Data », qu'il connaît bien, et visite uniquement la branche « Statistics ». Cependant, il ne sait pas où est la fonctionnalité recherchée et navigue à la fois dans « Summary » et dans « Tables ».

Utilisateur expérimenté qui recherche une commande

Comme dans le cas précédent, l'utilisateur identifie rapidement les branches candidates en haut de la hiérarchie grâce à son expertise. De plus, comme il connaît le nom de la commande, il peut se laisser guider par le titre des sous-menus lorsqu'il se trouve à proximité de celle-ci (cas b). Cependant, il risque toujours de naviguer dans la partie intermédiaire du menu. En effet,

celle-ci est moins fréquemment visitée que les parties hautes de la hiérarchie. De plus, les titres des sous-menus ne sont pas suffisamment concrets pour guider la navigation. C'est le cas d) de la Figure 3 où l'utilisateur est familier avec le haut et le bas de la hiérarchie car il connaît le nom de la commande recherchée.

L'étudiant est habitué à Stata et souhaite rechercher la commande « tabstat ». Il va directement dans « Statistics » car il sait qu'elle ne peut pas être dans « Data ». Lorsqu'il voit le titre « tables », il fait le rapprochement avec la commande « tabstat ».

Dans les catégories de la Figure 3, il est à noter que l'utilisateur est moins familier avec le niveau de profondeur intermédiaire. Ce point doit être pris en compte lors de la conception d'un menu hiérarchique. Nous revenons sur ce point dans le chapitre 7.

Nous retenons que la tâche de recherche peut imposer à tous les types d'utilisateurs d'emprunter des chemins détournés. Il est donc souhaitable de : 1) faciliter la remontée dans la hiérarchie lorsque l'utilisateur se trouve dans une mauvaise branche, et, 2) de fournir le contexte afin d'éviter que l'utilisateur ne se perde lorsque l'arbre est profond.



Nous retenons que la recherche d'une commande ou d'une fonctionnalité peut impliquer une exploration importante du système de menus. Il est souhaitable de faciliter la remontée rapide dans la hiérarchie et de fournir le contexte (position courante) à l'utilisateur.

Nous venons de présenter les deux tâches de navigation : le *feuillete* et la *recherche*. Nous présentons maintenant les différentes stratégies de recherche visuelle que peut adopter l'utilisateur.

1.4 Recherche visuelle

La recherche visuelle est un élément important dans le processus d'activation d'une commande. Plus l'utilisateur navigue dans un système de menus, plus le temps de recherche visuelle est conséquent par rapport au temps de sélection (section 1 du chapitre 3). Pour cette raison, il est important de bien comprendre les stratégies utilisées afin d'identifier les solutions interactionnelles facilitant la recherche visuelle et donc l'exploration dans les menus.

Nous considérons les 3 stratégies pour la recherche visuelle de [Norman 91] : la lecture, le balayage et le placement. La dernière se distingue des deux autres, car elle ne concerne que les utilisateurs très expérimentés. Nous détaillons chacune d'entre elles.

Lecture

Cette stratégie est la plus intuitive. Elle consiste à lire successivement chaque item afin de trouver la commande désirée sans risquer de la manquer : c'est une « inspection en série » des items (Figure 6). Elle est utilisée lorsque l'utilisateur n'a pas de connaissances a priori sur la position ou le nom de la commande. Elle est lourde cognitivement car pour chaque item, l'utilisateur doit : 1) le lire, 2) le comprendre et 3) évaluer sa pertinence par rapport à un but. Norman [Norman 91] distingue deux règles permettant de déterminer l'arrêt de la lecture (Figure 5) : la règle d'*optimisation* et la règle de *satisfaction*.

- La règle d'optimisation consiste à sélectionner l'item qui semble le plus en adéquation avec le but de l'utilisateur. Elle minimise les erreurs mais force l'utilisateur à lire (ou relire) tous les items avant d'en sélectionner un.
- La règle de satisfaction consiste à sélectionner le premier item que l'utilisateur juge correspondre à son but (évaluation supérieure à un seuil prédéterminé [Simon 76]). Cette solution, plus rapide, ne garantit pas un choix optimal et peut donc impliquer des chemins détournés.

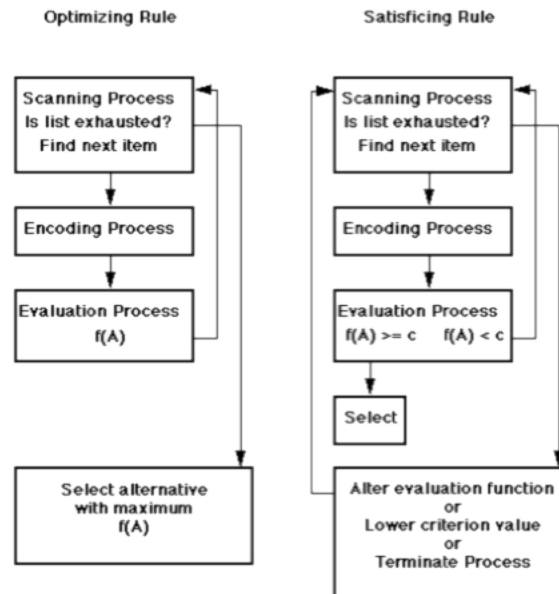


Figure 5 : Deux règles d'arrêt dans une stratégie de lecture : la règle d'optimisation (gauche) et la règle de satisfaction (droite). Illustration extraite de [Norman 91].

Lorsque les utilisateurs connaissent le nom de la commande cible, ils peuvent utiliser une stratégie de lecture, mais utilisent généralement une stratégie de balayage plus rapide.

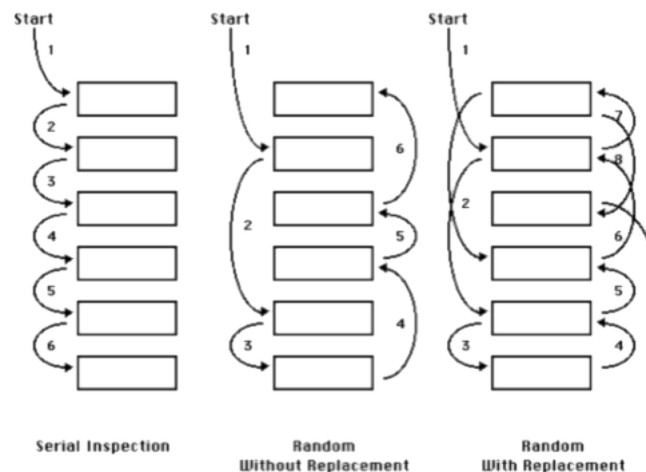


Figure 6 : Les 3 types d'inspection. Inspection en série, inspection aléatoire sans remplacement, inspection aléatoire avec remplacement. Illustration extraite de [Norman 91].

Balayage

Cette stratégie est utilisée lorsque l'utilisateur connaît le nom mais pas la position de la commande recherchée. Elle consiste à survoler les items sans réellement les lire. L'utilisateur fait simplement une correspondance visuelle entre le libellé de l'item courant et le nom de la commande recherchée (« visual matching process ») [Norman 91]. C'est la différence entre la

reconnaissance globale (holistique) et la reconnaissance analytique des mots [Taylor 76]. Le temps de fixation du regard est beaucoup plus court que dans le cas de la stratégie de lecture et l'utilisateur peut facilement faire des erreurs d'évaluation. Nous distinguons 4 cas (Figure 7) :

- *détection positive (hit)* : l'utilisateur fait une correspondance et les éléments sont effectivement identiques.
- *détection d'un faux positif (false positive)* : l'utilisateur fait une correspondance alors que les éléments sont différents.
- *détection manquée (miss)* : l'utilisateur fait une non-correspondance alors que les éléments sont identiques.
- *rejet correct (correct rejection)* : l'utilisateur fait une non-correspondance et les éléments sont effectivement différents.

Deux types de parcours visuel ont été observés pour le balayage (Figure 6). Le premier consiste à sauter des items sans jamais fixer deux fois le même item en gardant une trace des éléments déjà visités (*inspection aléatoire sans remplacement*). Le second, moins efficace, consiste toujours à sauter des items mais le regard peut refixer un élément déjà visité (*inspection aléatoire avec remplacement*).

Nous retenons que le balayage est beaucoup plus rapide que la lecture mais qu'il peut introduire des erreurs (par exemple manquer un item en le balayant trop vite du regard).

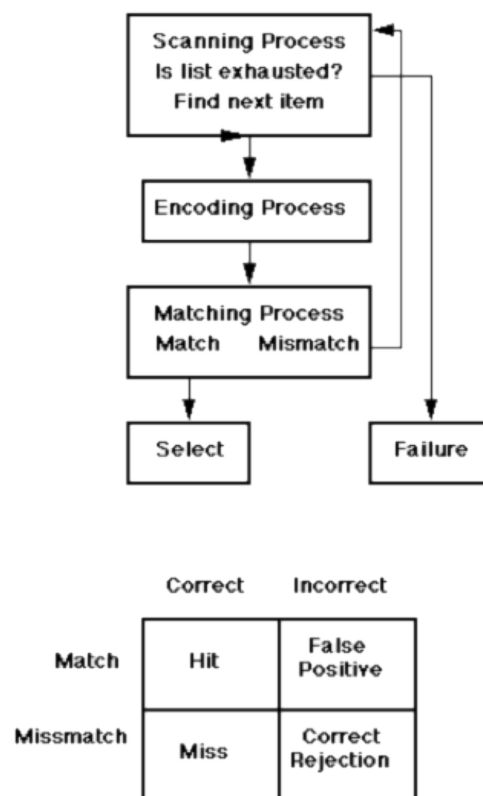


Figure 7 : Processus de correspondance entre le nom de la commande recherchée et le libellé de l'item balayé. Illustration extraite de [Norman 91].

Placement

Le « placement » consiste à placer quasiment instantanément le regard sur l'élément recherché. Cette stratégie est possible lorsque l'utilisateur expérimenté connaît la position exacte de la commande recherchée. Des attributs visuels comme les séparateurs ou les icônes (section 1.2 du chapitre 3) facilitent le « placement » car ils fournissent des repères visuels. Le

placement s'apparente plus à de la prise de décision (section 1 du chapitre 2) qu'à de la recherche visuelle.



Nous retenons que les utilisateurs lisent séquentiellement les items lorsque le nom de la commande n'est pas connu ou les balayent dans le cas contraire. La seconde stratégie est beaucoup plus rapide car elle ne met en jeu qu'un processus de correspondance visuelle entre le libellé de l'item visualisé et le nom de la commande recherché.

Nous présentons maintenant les solutions conceptuelles favorisant l'exploration dans les systèmes de menus. En particulier, nous montrons comment la propriété de prévisualisation aide le balayage dans les sous-menus.

1.5 Solutions conceptuelles pour l'exploration

De notre étude de l'exploration dans les systèmes de menus, nous retenons que quelle que soit la tâche, l'utilisateur est souvent amené à visiter un nombre important de sous-menus. Nous détaillons ici trois solutions qui favorisent l'exploration dans les systèmes de menus de notre espace MenUA que nous avons déjà caractérisé au chapitre 4, sections 1.2 et 1.3 (critère : *rapidité et précision*, aspect : *recherche visuelle et sélection d'un item*). Ces trois solutions sont :

- la prévisualisation, qui facilite la recherche visuelle dans les sous-menus ;
- l'observabilité du chemin parcouru, qui aide l'utilisateur à ne pas se perdre dans une structure profonde ;
- l'accès direct aux parents, qui permet de remonter rapidement dans la hiérarchie.

Ces solutions ont motivé la conception des Wave menus présentées dans la section 2.

La prévisualisation

La « prévisualisation » consiste à afficher automatiquement le contenu d'un sous-menu associé à un item qui a le focus. Ceci permet à l'utilisateur de balayer rapidement plusieurs sous-menus en les visualisant continûment et successivement. La prévisualisation est un retour visuel proactif [Rekimoto 03] car elle montre le contenu du sous-menu avant même que l'utilisateur ait besoin de l'activer. La Figure 8, inspirée de [Furnas 97b], montre qu'en plus des items du menu courant, l'utilisateur peut accéder aux items du sous-menu (car le sous-menu a été ouvert automatiquement).

La prévisualisation réduit la charge cognitive de l'utilisateur en lui indiquant à l'avance la pertinence du contenu du sous-menu [Chigona 02]. Elle augmente l'information disponible et évite ainsi de rentrer dans un sous-menu inutilement.

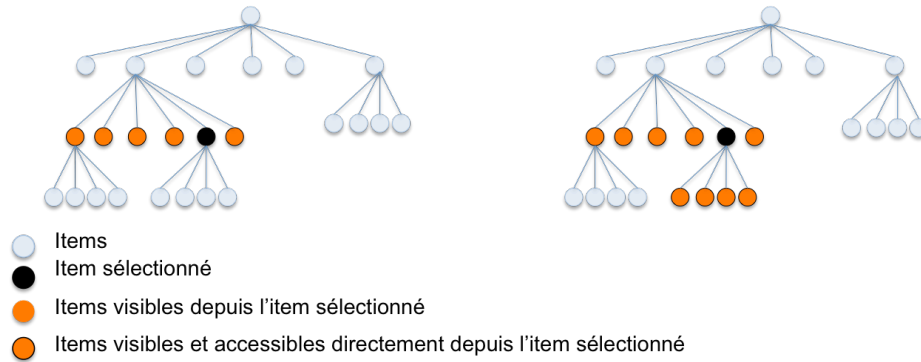


Figure 8 : A gauche, un menu hiérarchique sans prévisualisation. L'utilisateur peut sélectionner seulement les items du menu courant. Pour sélectionner les items du sous-menu, il doit explicitement demander l'ouverture du sous-menu. A droite, un menu hiérarchique avec prévisualisation. L'utilisateur peut sélectionner les items du sous-menu car le sous-menu a été ouvert automatiquement.

La prévisualisation, qui est une solution utilisée dans de nombreux menus, est fort utile pour la navigation en largeur. Un simple geste continu permet d'avoir un aperçu rapide de tous les sous-menus. Par exemple, la Figure 9 illustre le cas d'un utilisateur balayant trois sous-menus d'un Compound Marking menu avec un mouvement circulaire.

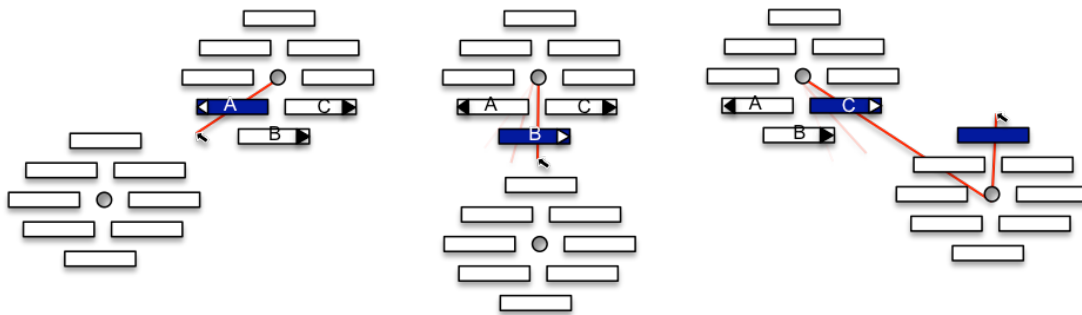


Figure 9 : Exploration en largeur de 3 sous-menus avec un Compound Marking menu. Grâce à la prévisualisation, l'utilisateur peut visiter 3 sous-menus continûment sans clic souris.

La Figure 10 illustre le même parcours visuel que précédemment, mais avec des Multi-Stroke menus qui ne possèdent pas la propriété de prévisualisation. L'utilisateur est forcé d'ouvrir chaque sous-menu afin de pouvoir le balayer du regard. Si la commande ne s'y trouve pas, il doit de plus aller pointer dans chacune des zones centrales pour fermer le sous-menu. Il doit ensuite recommencer les mêmes opérations pour balayer le sous-menu suivant et ainsi de suite. En diminuant le nombre d'actions nécessaires pour balayer les sous-menus, la prévisualisation permet à l'utilisateur de se concentrer sur la recherche visuelle.

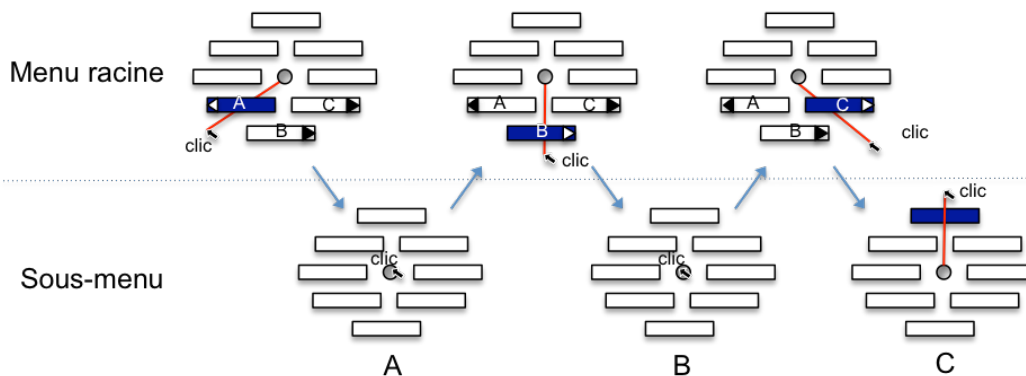


Figure 10 : Exploration en largeur de 3 sous-menus avec un Multi-Stroke menu (qui n'a pas la propriété de prévisualisation). L'utilisateur doit : 1) explicitement demander l'ouverture d'un sous menu, 2) le parcourir

visuellement, 3) remonter au menu racine (en cliquant au centre) si la commande recherchée ne s'y trouve pas et ainsi de suite.

La prévisualisation favorise donc la tâche de *feuilletage* en facilitant l'exploration en largeur et la tâche de *recherche* en évitant à l'utilisateur de descendre/remonter inutilement dans les sous-menus.

Bien qu'elle soit importante, la prévisualisation n'a pas reçu beaucoup d'attention dans la littérature sur les menus [Forsström 06]. Peut être parce que cette technologie est présente depuis si longtemps dans les menus linéaires que de nombreux utilisateurs et auteurs ne se rendent plus compte de son existence et par là même de son importance (sans compter que la plupart des travaux de recherche ont quelque peu délaissé le mode novice). Ceci explique sans doute que de récentes techniques ne permettent plus la prévisualisation [Kobayashi 03, Maya, Zhao 04, Zhao 06]. Une contribution de ce travail est de réhabiliter la solution de prévisualisation.

Observabilité du chemin parcouru

L'observabilité du chemin parcouru consiste à afficher le chemin déjà traversé par l'utilisateur et lui indique l'endroit où il se trouve dans la hiérarchie (Figure 11-gauche), comme nous l'avons dit dans la section 2.3 du chapitre 4.

La navigation dans un système de menus peut s'avérer une charge cognitive importante pour l'utilisateur car il doit lire un grand nombre de libellés. Après avoir visité plusieurs sous-menus, il peut facilement oublier leurs titres et les revisiter inutilement. En affichant le chemin déjà parcouru, l'utilisateur n'est pas forcé de mémoriser les sous-menus traversés et peut éviter de les visiter une seconde fois. Cette solution réduit donc l'effort cognitif de l'utilisateur et la quantité d'exploration. De plus, elle favorise l'apprentissage de l'organisation (section 3 du chapitre 4) en affichant une commande dans son contexte.

Les menus linéaires affichent le chemin parcouru. Ils affichent, non seulement, les items qui permettent d'accéder au menu courant (en surbrillance), mais également les menus parents. A l'opposé, certains menus circulaires, comme les Multi-Stroke menus, n'affichent pas le chemin parcouru car les sous-menus sont superposés et masquent donc les menus supérieurs.

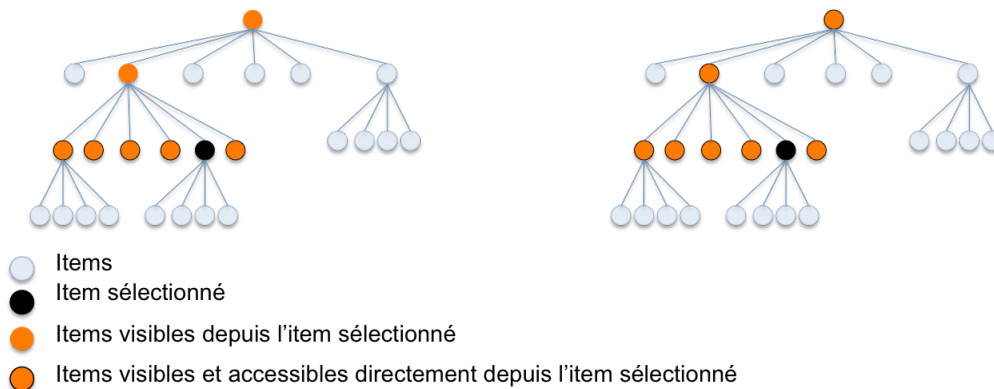


Figure 11: A gauche, un menu hiérarchique dont le chemin parcouru est observable. Le chemin déjà parcouru dans la branche courante est affiché mais les items ne peuvent pas être sélectionnés. A droite, un menu hiérarchique qui permet d'accéder directement aux parents.

Accès direct aux parents

L'« accès direct aux parents » consiste à pouvoir remonter plusieurs niveaux de la hiérarchie en une seule action (Figure 11-droite). Dans le cas où l'utilisateur a inutilement parcouru plusieurs niveaux de la hiérarchie, l'accès direct aux parents permet de revenir, en une action, à l'embranchement où l'utilisateur s'est trompé et évite de remonter pas à pas dans la hiérarchie. Elle repose sur la solution précédente (observabilité du chemin parcouru) ; Tous les

menus parents devant être visibles afin que l'utilisateur puisse les atteindre directement. Les menus linéaires permettent à tout moment de sélectionner très facilement n'importe quel item de la branche courante.

L'accès direct aux parents est surtout utile pour l'exploration en profondeur. Elle favorise la tâche de recherche lorsque la hiérarchie comporte au moins 3 niveaux de profondeur.



Nous retenons trois solutions conceptuelles améliorant la navigation dans les systèmes de menus :

- la prévisualisation ;
- l'observabilité du chemin parcouru ;
- l'accès direct aux parents.

Ces solutions conceptuelles ne sont pas limitées aux systèmes de menus et peuvent être généralisées aux systèmes d'information tels que le Web [Furnas 97b]. Ainsi, le chemin déjà parcouru sur le Web n'est généralement pas affiché et l'utilisateur dispose de très peu d'informations sur la page accessible à partir d'un lien hypertexte. « Snap shots » [Snapshots] est un outil qui permet d'avoir un aperçu de la page associée à un lien hypertexte. Cette page apparaît dans une bulle d'info avant que l'utilisateur clique dessus. L'outil coolpreview [coolpreview] permet en plus d'interagir directement avec la page à travers la bulle d'info.

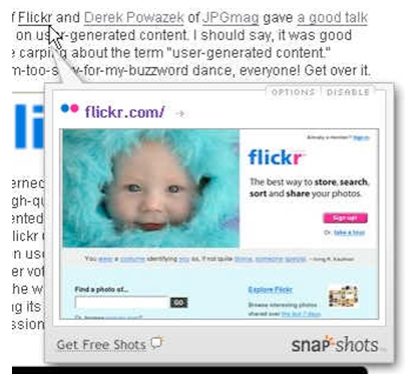


Figure 12 : Outil « Snap shots » permet d'avoir un aperçu de la page accessible à partir d'un lien hypertexte. Illustration extraite de [Snapshots].

Curieusement, ces solutions n'ont pas suscité beaucoup d'attention dans la littérature sur les menus. Ceci est sans doute lié au fait que la plupart des études récentes ont été consacrées à l'amélioration du critère *rapidité et précision* en mode expert. Nous pensons que l'amélioration du mode novice est un problème tout aussi important étant donné que c'est celui qui est le plus utilisé en pratique (même par des utilisateurs experts). C'est pourquoi nous utilisons ces connaissances sur la navigation pour améliorer le mode novice des Marking menus. A partir des trois solutions conceptuelles rappelées dans cette section, nous proposons une nouvelle technique de menus.

2 Wave menus

Cette section décrit une nouvelle technique de menus appelée Wave menus conçue pour améliorer l'efficacité des Marking menus hiérarchiques en mode novice. Après un rappel des limitations des Marking menus hiérarchiques, nous exposons le principe de fonctionnement des Wave menus puis nous détaillons les propriétés interactionnelles au regard de notre espace MenUA.

2.1 Rappel des limitations des Marking menus hiérarchiques

Les Marking menus soulèvent des problèmes à la fois en mode novice et en mode expert (chapitres 3 et 4).

Les Compound Marking menus (CMM), illustrés Figure 13, sont imprécis en mode expert [Kurtenbach 93, Zhao 04] lorsque le nombre de commandes est important car les utilisateurs ont des difficultés à composer spatialement plusieurs marques (en particulier sur les diagonales).

Certains gestes des CMM sont ambigus en mode expert [Zhao 04]. Les marques étant indépendantes à l'échelle, une trace peut parfois correspondre à plusieurs commandes. Pour éviter ces ambiguïtés, certains gestes ne sont pas utilisés. Le nombre total de commandes d'un CMM de profondeur 3 est ainsi limité à 400 commandes (au lieu de 512) [Zhao 04].

Les CMM nécessitent beaucoup d'espace écran en mode novice [Zhao 04] car les sous-menus sont affichés dans la direction de l'item parent (Figure 17). Un CMM de profondeur 3 a besoin de près de 4 fois plus d'espace sur l'axe horizontal qu'un menu linéaire.

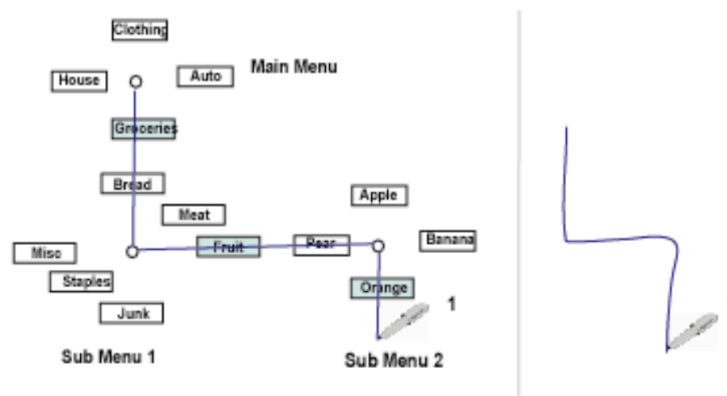


Figure 13 : Les Compound Marking menus (CMM). A gauche, le mode novice. A droite le mode expert. Illustration extraite de [Zhao 04].

Alors que la version originale des CMM vérifie les trois solutions présentées précédemment, certaines implémentations³ ne favorisent pas la navigation en mode novice. Par exemple, les CMM du logiciel de modélisation 3D Maya [Maya] ne disposent pas de la prévisualisation. Dans la suite de ce chapitre, nous considérons la version originale des CMM [Kurtenbach 93] car nous nous concentrons sur la navigation.

Les Multi-Stroke menus (MSM) [Zhao 04] améliorent l'efficacité des Marking menus hiérarchiques en mode expert (Figure 14). Ils augmentent la précision et évitent les ambiguïtés en composant les marques temporellement (au lieu de spatialement). De plus, les MSM nécessitent moins d'espace écran car les sous-menus sont superposés. Cependant, cette

³ Les concepteurs de Maya ont préféré conserver « l'indépendance à l'échelle » en mode novice laquelle n'était pas compatible avec la prévisualisation. Un mouvement circulaire de la souris ne permet pas d'avoir un aperçu des différents sous-menus.

disposition spatiale dégrade le mode novice car elle ne permet plus d'avoir les propriétés de prévisualisation, d'observabilité du chemin parcouru et d'accès direct aux parents.

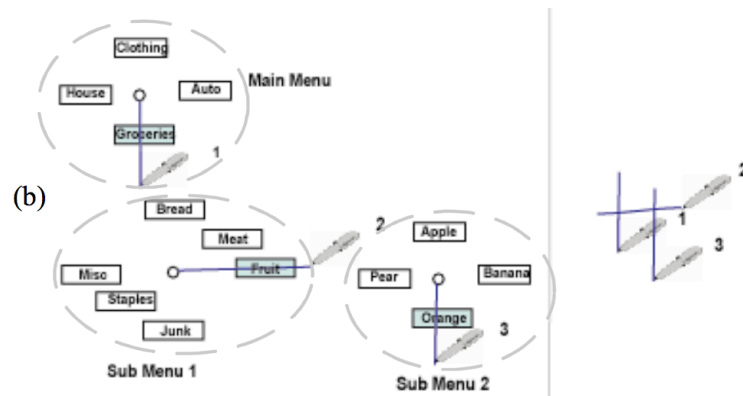


Figure 14 : Les Multi-Stroke menus (MSM). A gauche le mode novice, à droite, le mode expert. Illustration extraite de [Zhao 04].



Nous retenons que les Multi-Stroke menus sont plus efficaces que les Compound Marking Menus en mode expert mais leur mode novice est fortement dégradé.

2.2 Principe des Wave menus

Deux voies ont été jusqu'à présent explorées pour la disposition spatiale des menus hiérarchiques : la juxtaposition spatiale (menus linéaires, Compound Marking menus) et la superposition (Multi-Stroke menus). Les Wave menus reposent sur une nouvelle stratégie : une organisation concentrique « inversée » où chaque sous-menu apparaît sous la forme d'un anneau d'autant plus excentré qu'il est haut dans la hiérarchie du système de menus.

La Figure 15 illustre le fonctionnement d'un Wave menu à deux niveaux de profondeur. L'utilisateur presse le bouton de la souris et le menu apparaît sous la forme d'un anneau centré autour du curseur comme pour un Marking menu classique (état A). En dessinant une marque en direction de l'item souhaité (transition 1), l'anneau s'écarte progressivement comme si le curseur le poussait de l'intérieur vers l'extérieur. Le sous-menu de l'item correspondant apparaît alors progressivement de la zone centrale en fonction de la place disponible. Une fois qu'il a atteint sa taille définitive, l'anneau du menu parent (menu racine) se fige et entoure le sous-menu (Etat B). Durant l'effet d'élargissement, les anneaux ne se rétractent pas, même si le curseur revient en arrière.

Pour sélectionner un item dans le sous-menu, l'utilisateur fait une marque en direction de cet item (transition 2). Comme l'item est terminal (i.e. il ne donne pas accès à un sous-menu), l'anneau ne s'écarte pas. Après avoir relâché le bouton de la souris, la marque de confirmation apparaît pendant un délai de 100ms (Etat D).

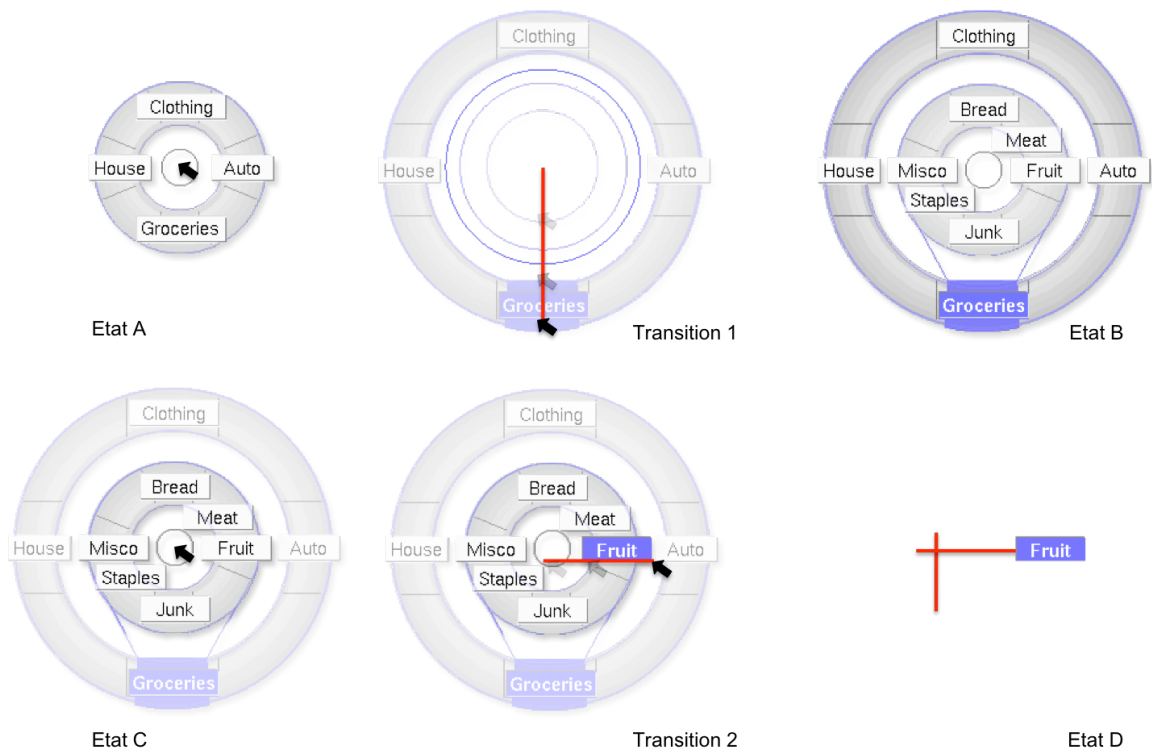


Figure 15 : Fonctionnement d'un Wave menu à deux niveaux de profondeur. (état A) Le menu apparaît centré autour du curseur. En dessinant une marque en direction de l'item (transition 1), le premier niveau de la hiérarchie s'élargit pour laisser apparaître progressivement le sous-menu (état B). En appuyant sur le sous-menu, celui-ci prend le focus (Etat C). Avec une seconde marque (transition 2), l'utilisateur active l'item. Une confirmation rappelle pendant 300ms, l'item activé (Etat D).

Le même mécanisme est utilisé si le système de menus a plus de deux niveaux de profondeur (Figure 16) : tous les menus parents s'élargissent et le sous-menu apparaît progressivement de la zone centrale. Les menus s'écartent donc progressivement du point d'activation. Cet effet rappelle la propagation de vagues. Le menu le plus excentré est le menu le plus haut dans la hiérarchie (menu racine), alors que le menu le plus proche du point d'activation est le menu le plus profond de la branche visible.

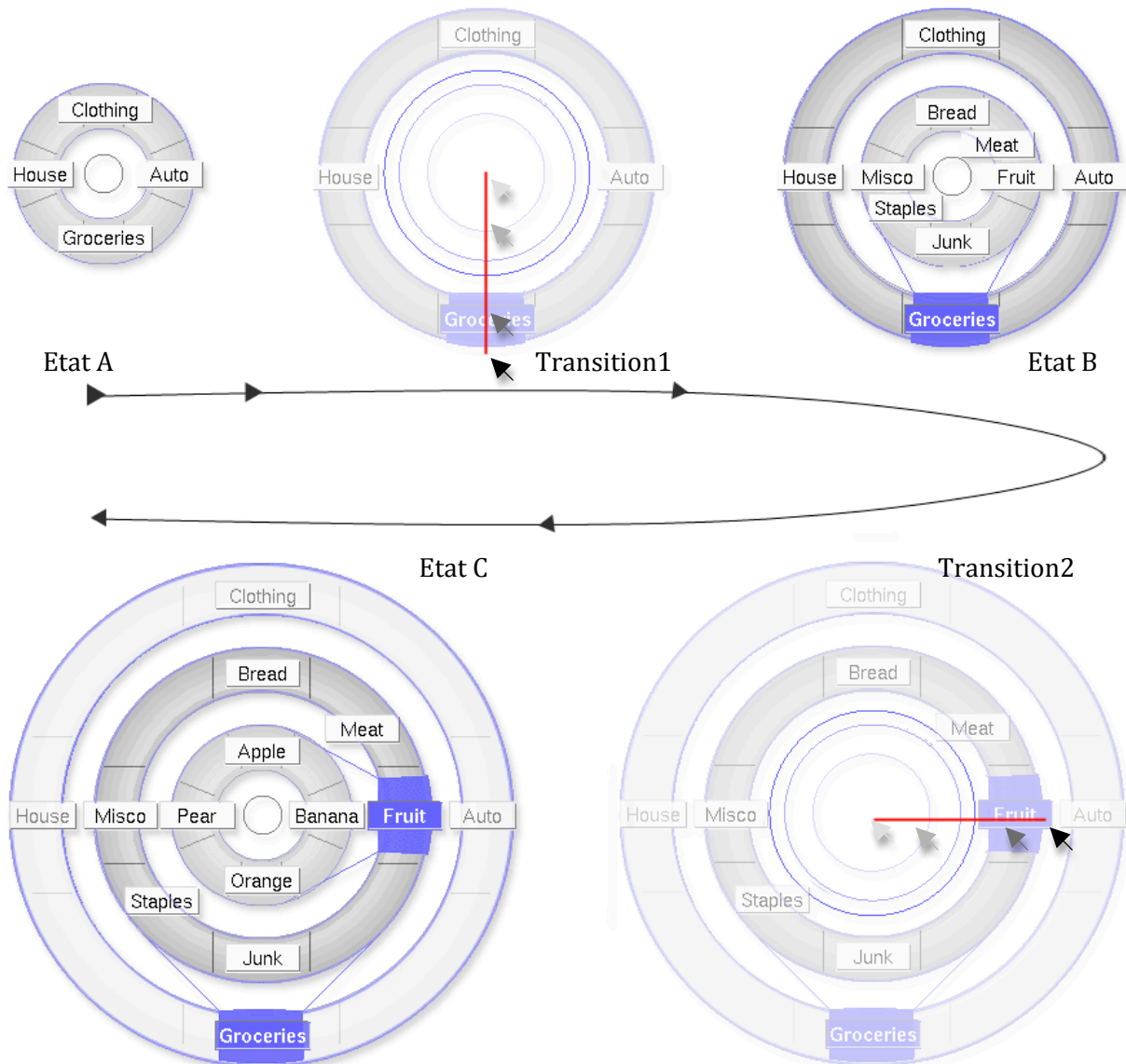


Figure 16 : Fonctionnement d'un Wave menu à trois niveaux. (état A) Le menu apparaît centré autour du curseur. En dessinant une marque en direction de l'item (transition 1), le premier niveau de la hiérarchie s'élargit pour laisser apparaître progressivement le sous-menu (état B). Avec une seconde marque (transition 2), le troisième niveau apparaît au centre et les deux premiers niveaux s'élargissent (état C). Cet effet rappelle la propagation des vagues.

Enfin, en mode expert, Les Wave menus fonctionnent comme les Multi-Stroke menus : l'utilisateur active une commande en réalisant une série de marques simples en relâchant le bouton de la souris à chaque fois (composition temporelle).

2.3 Propriétés des Wave menus

Nous détaillons maintenant les propriétés des Wave menus au regard de notre espace MenUA. Nous présentons d'abord les propriétés associées aux critères d'utilisabilité puis nous décrivons les critères d'applicabilité.

2.3.1 Rapidité et précision

Activation du menu

Le Wave menu s'active, comme tous les Marking menus, lorsque l'utilisateur appuie sur l'un des boutons de la souris.

Recherche visuelle

Nous avons vu en détail dans la section 1, l'importance de la prévisualisation pour améliorer la recherche visuelle dans les sous-menus. Le Wave menu possède cette propriété. L'organisation concentrique permet en effet de se déplacer sur les items de n'importe quel menu de la hiérarchie et de faire apparaître implicitement les sous-menus correspondant une fois le menu élargit. Un mouvement circulaire sur l'anneau extérieur permet d'avoir un aperçu rapide du contenu des différents sous-menus. Cette organisation évite de superposer les menus, comme dans le cas des Multi-Stroke menus.

Une propriété intéressante de cette technique est que les sous-menus sont toujours affichés à la même place, autour du point d'activation. Ceci diffère des menus linéaires et des Compound Marking menus. Les sous-menus sont affichés à gauche ou droite dans les menus linéaires. Ils sont affichés dans la direction de l'item parent dans les Compound Marking menus, ce qui peut nécessiter un déplacement du regard important pendant la recherche visuelle (Figure 17). Au contraire, les Wave menus ne forcent pas à déplacer le regard car la position des sous-menus est toujours la même. L'utilisateur peut conserver son focus d'attention au centre du menu pendant toute la durée de la recherche visuelle.

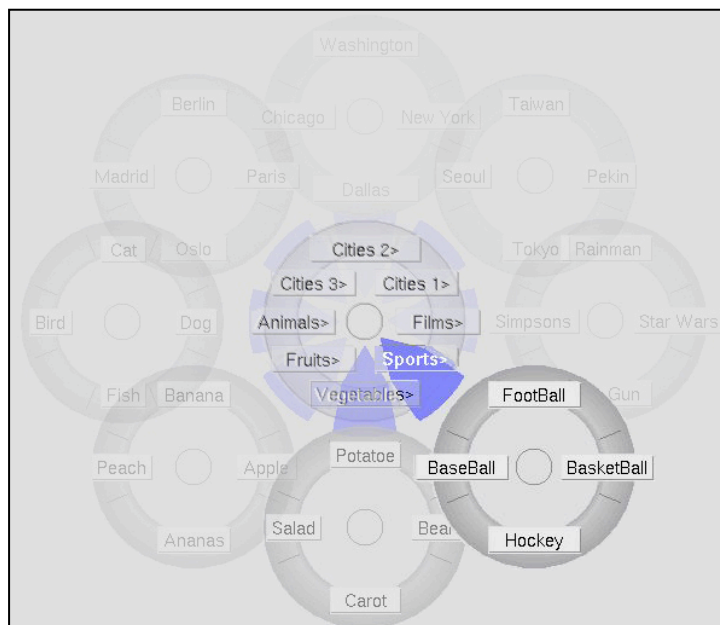


Figure 17 : Parcours visuel important durant la recherche visuelle dans les CMM.

Sélection d'un item

L'indépendance à l'échelle est une propriété qui favorise la sélection de commandes. Les gestes des Wave menus sont indépendants à l'échelle en mode expert (car les Wave menus fonctionnent alors comme les Multi-Stroke menus). L'utilisateur peut également faire des marques (« dragging state ») de n'importe quelle taille en mode novice. Seule l'orientation est utilisée pour sélectionner un item. Si, l'utilisateur relâche le bouton de la souris avant d'avoir atteint le libellé de l'item (Figure 18-b), une rapide animation est jouée pour continuer l'élargissement du menu parent et laisser apparaître progressivement le sous-menu. Afin de permettre à l'utilisateur de dépasser le libellé de l'item, les menus parents sont inactifs pendant l'exécution de la marque (Figure 18-b-c-d). Ainsi l'utilisateur peut faire de longues marques sans risquer de sélectionner un item parent accidentellement. C'est une différence avec les Compound Marking menus qui n'ont pas la propriété d'indépendance à l'échelle en mode novice. En effet, comme les sous-menus des Compound Marking menus sont affichés dans la direction de l'item, une marque trop longue risque de rentrer dans le sous-menu et d'en activer un item par erreur (Rappelons que l'implémentation des Compound Marking menus dans le logiciel

Maya permet des marques indépendantes à l'échelle en mode novice, mais du coup ne permet plus la prévisualisation). La représentation spatiale des Compound Marking menus ne permet donc pas d'avoir à la fois l'indépendance à l'échelle et la prévisualisation contrairement à la représentation concentrique inversée des Wave menus.

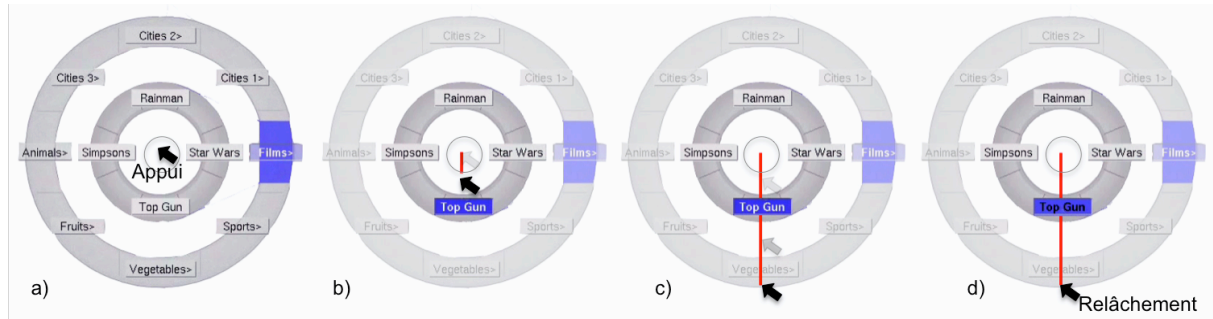


Figure 18 : Les gestes sont indépendants de l'échelle en mode novice. Pour sélectionner une commande, l'utilisateur peut relâcher le bouton avant d'avoir atteint l'item (b) ou après l'avoir dépassé (d). Pendant l'interaction, les menus parents sont inactifs (semi-transparents) pour ne pas être sélectionnés par erreur.

Lorsque l'utilisateur ne dessine pas de marque (« Tracking state »), il peut accéder à tous les menus parents (*accès direct aux menus parents*). Ces derniers peuvent être sélectionnés par un simple clic. L'utilisateur peut donc remonter très rapidement dans la hiérarchie (section 1). Afin de rester cohérent avec les Marking menus, il est toujours possible de remonter dans la hiérarchie en cliquant à l'intérieur du cercle central. Une rapide animation est alors jouée montrant le menu qui se rétracte (Figure 19).

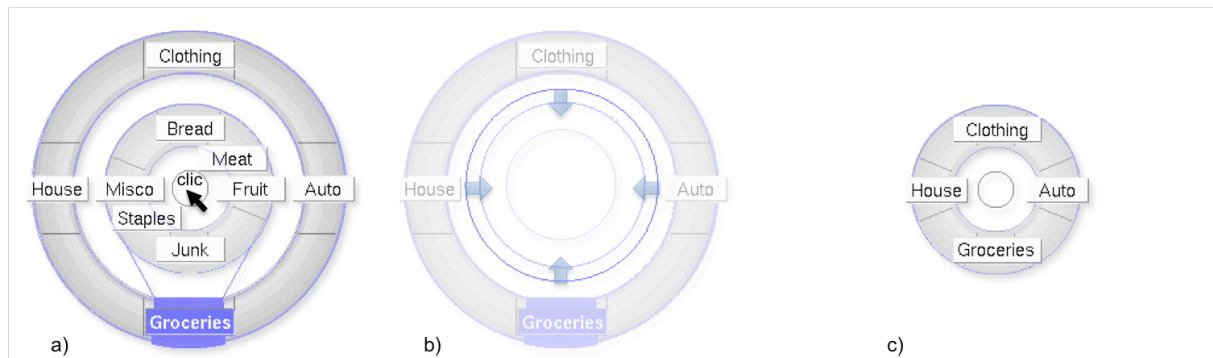


Figure 19 : Remontée dans la hiérarchie avec les Wave menus. L'utilisateur clique dans la zone centrale (a). Une rapide animation est jouée montrant le menu parent qui se rétracte (b).

Ces propriétés du Wave menu concourent à améliorer la navigation (recherche visuelle et la sélection d'un item) en mode novice. L'expérience, présentée dans la section 3, confirme expérimentalement que le Wave menu favorise la navigation en mode novice.

Mode expert

Comme nous l'avons dit précédemment, le mode expert des Wave menus fonctionne comme celui des Multi-Stroke menus. Les marques sont composées temporellement afin de garantir un taux d'erreur faible même si le menu comporte plus de 2 niveaux de profondeur.

2.3.2 Apprentissage et Mémorisation

Apprentissage de la technique

L'originalité de la représentation spatiale (concentrique inversée) pouvait être un handicap à la prise en main de la technique. Aussi, nous avons demandé à des utilisateurs⁴ d'essayer la technique sans aucune information. Ces deniers ont réussi à utiliser la technique sans aucune explication préalable. De plus pour favoriser la prise en main, nous avons d'une part utilisé un effet d'élargissement contrôlé intégralement par l'utilisateur. Cet effet permet de mieux assimiler que les menus parents s'éloignent. Si cet effet avait été contrôlé par le système, l'animation aurait pu être soit trop courte (pour les utilisateurs inexpérimentés) soit trop longue (pour les utilisateurs expérimentés). Cet effet fonctionne ainsi : La distance du curseur par rapport au centre du menu détermine le rayon de l'anneau, comme si le curseur « poussait » l'anneau de l'intérieur. Les utilisateurs comprennent ainsi rapidement le fonctionnement de la technique. D'autre part, lorsque le menu est affiché, les utilisateurs peuvent cliquer directement sur les items visibles afin que le menu reste compatible avec un fonctionnement classique de sélection à la souris. Lorsque que le bouton de la souris est déjà enfoncé, l'utilisateur peut activer un item d'un sous menu en se déplaçant dessus puis en relâchant la souris en mode novice. Les utilisateurs peuvent donc choisir d'interagir comme ils le font usuellement.

Apprentissage et mémorisation de l'organisation des commandes

Comme pour les Marking menus, l'organisation circulaire des Wave menus permet d'associer facilement une commande à une orientation (section 2.3 du chapitre 4). De plus, les menus parents sont visibles et le chemin parcouru est mis en surbrillance. L'utilisateur peut facilement situer un item dans son contexte, identifier sa profondeur et retrouver le chemin pour y accéder.

Apprentissage et mémorisation du mode expert

L'utilisateur exécute les mêmes gestes en mode novice et en mode expert comme avec les Marking menus. Ceci permet à l'utilisateur d'apprendre progressivement à sélectionner des commandes en mode expert sans changer de technique d'interaction. Les Wave menus conservent ainsi une transition fluide du mode novice au mode expert.

2.3.3 Satisfaction

Le fait que le Wave menu soit complètement contrôlé par l'utilisateur peut concourir à la satisfaction de l'utilisateur. Le menu apparaît comme malléable et pouvant être étiré pour voir ce qu'il y a à l'intérieur.



Nous retenons pour les critères d'utilisabilité que les Wave menus permettent de prévisualiser les sous-menus, d'observer le chemin déjà parcouru et d'accéder directement aux menus parents.

⁴ Les 12 utilisateurs, qui ont participé à notre expérience décrite section 3, devaient préalablement découvrir par eux même le fonctionnement de la technique. Les résultats sont détaillés dans la section 3.5.

Après les critères d'utilisabilité (chapitre 4), nous abordons les critères d'applicabilité de MenUA (chapitre 7)

2.3.4 Adéquation à l'application

Les Wave menus permettent d'effectuer un grand nombre de commandes. Comme le mode expert est identique à celui des Multi-Stroke menus, il n'existe pas de gestes ambigus et le taux d'erreur est faible. Les Wave menus sont donc utilisables avec $8*8*8 = 512$ commandes.

Cependant, les Wave menus ne résolvent pas le problème du nombre de commandes en largeur. Comme les Multi-Stroke menus, ils sont limités à 8 items par sous-menu. Ce problème fait l'objet du chapitre suivant.

2.3.5 Adéquation à la plateforme

Les Wave menus peuvent être utilisés avec tous types de dispositifs de pointage 2D (absolus ou relatifs). Comme les Multi-Stroke menus, ils restent utilisables sur de petites surfaces d'entrée comme un touchpad, car l'utilisateur peut faire des marques superposées pour descendre dans la hiérarchie (Figure 20). A l'opposé, les Compound Marking menus nécessitent des marques plus longues (en particulier en mode novice) qui peuvent être incompatibles avec de telles surfaces.

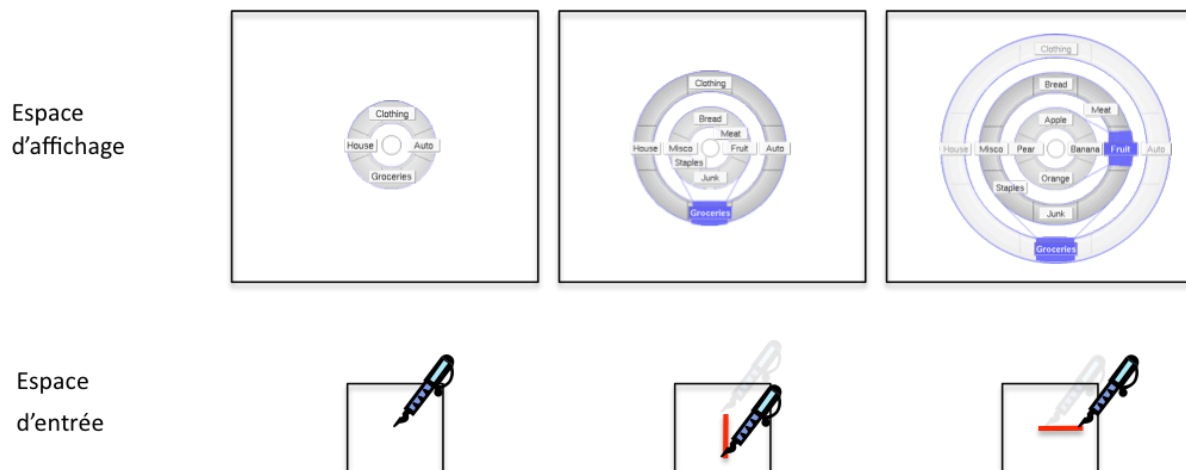


Figure 20 : Surface d'entrée minimale pour utiliser un Wave menu quelque soit sa profondeur. L'utilisateur peut réaliser des marques superposées pour descendre progressivement dans la hiérarchie.

Bien que les Wave menus affichent davantage d'informations que les Multi-Stroke menus, la taille écran minimale pour qu'ils puissent fonctionner est la même (Figure 21). Ils restent en effet utilisables lorsque les anneaux extérieurs ne sont plus accessibles, l'utilisateur se contentant alors d'interagir avec le menu situé au centre de la représentation, c'est-à-dire le menu le plus profond de la branche actuellement explorée. Certaines fonctionnalités ne sont plus disponibles (la prévisualisation, l'accès direct aux parents), mais le menu peut toujours être utilisé en mode "dégradé" qui est en fait identique au mode "normal" des Multi-Stroke menus. C'est une différence essentielle avec les Compound Marking menus qui ne sont pas utilisables près des bords de l'écran.

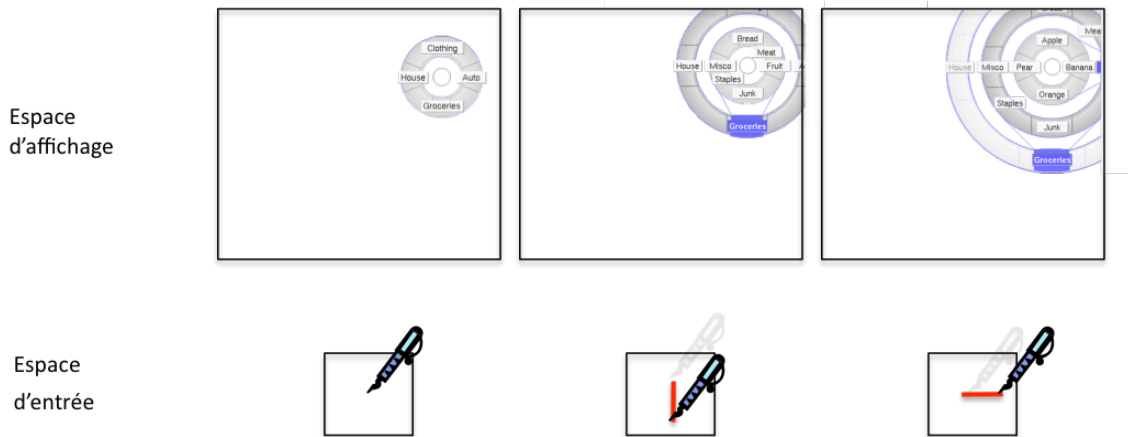


Figure 21 : Les Wave menus sont utilisables près des bords de l'écran en mode dégradé (la prévisualisation ou l'accès direct aux parents ne sont plus possibles).

Contrairement aux Compound Marking menus, les Wave menus peuvent donc être utilisés à la fois sur de petites surfaces d'entrée et de sortie comme les Multi-Stroke menus. Mais à la différence des Multi-Stroke menus, il est possible de prévisualiser ou d'accéder aux parents dès qu'il y a suffisamment d'espace écran. Les Wave menus résistent donc à différents contextes d'utilisation tout en exploitant au mieux l'espace disponible. Ceci est possible car l'organisation concentrique inversée relègue à l'extérieur les informations les moins utiles et conserve au centre, les menus indispensables à l'interaction.

2.3.6 Adéquation à la tâche

Transitions

Les Wave menus peuvent être utilisés en guise de menus contextuels. Ils favorisent la transition Objet-Commande grâce à la propriété « Sur place ».

Une autre propriété des Wave menus est qu'ils sont efficaces pour la transition Commande-Objet (retour à l'objet d'intérêt après l'activation d'un item). Comme nous l'expliquions précédemment, l'utilisateur peut descendre dans la hiérarchie en interagissant uniquement avec le menu au centre de la représentation. Le curseur reste donc proche du point d'activation. La transition pour revenir à l'objet d'intérêt sera donc courte et rapide.

Les Wave menus sont compatibles avec le contrôle de valeurs (Transition Commande-Valeur) par « crossing ». Si un item est associé à une « commande contrôlable », l'utilisateur doit traverser l'anneau extérieur pour commencer le contrôle continu de la valeur de cette commande. Le menu s'efface alors pour laisser l'utilisateur se concentrer sur la manipulation.

Contexte visuel

Plusieurs représentations visuelles ont été testées pour le Wave menu (Figure 22). La première alternative consiste à utiliser des disques pleins (b). Elle a l'avantage de mettre en valeur la hiérarchie, mais elle a l'inconvénient d'occulter fortement le contexte de l'utilisateur (surtout lorsque le système de menus est complètement ouvert). La transparence pourrait aider à augmenter le contexte visuel.

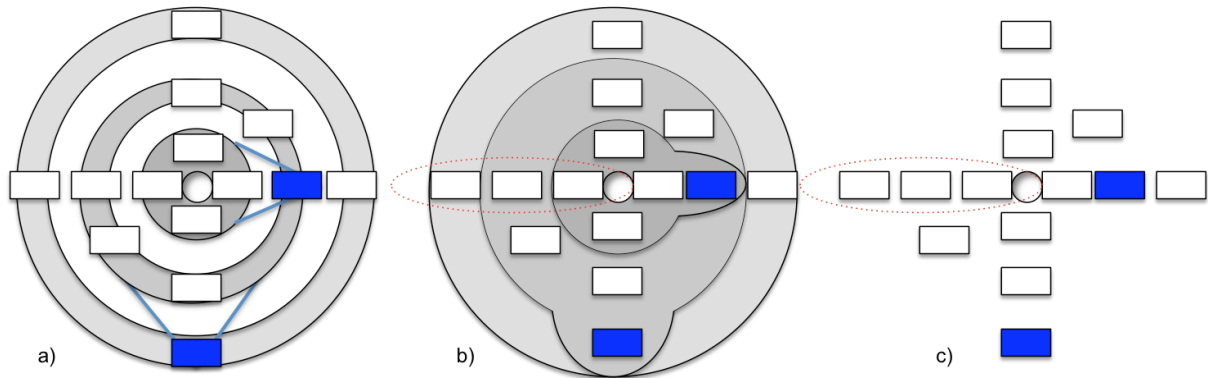
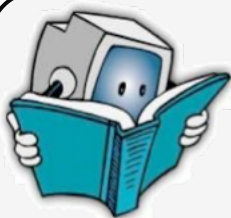


Figure 22 : Représentations du Wave menu. (a) la représentation adoptée. (b) une représentation qui met en valeur la hiérarchie, mais qui est occulte fortement le contexte. (c) une représentation qui maximise le contexte visuel, mais qui ne permet pas de comprendre la hiérarchie.

La seconde alternative (c) consiste à afficher uniquement les labels, comme proposée dans [Tapia 95], pour réduire l'occupation spatiale. Elle maximise le contexte visuel, mais la hiérarchie n'est plus du tout perceptible. De plus, elle introduit des associations visuelles entre les items qui n'ont pas de raison d'être. Par exemple, les trois items entourés en pointillé (Figure 22-c) peuvent être perçus comme appartenant à un même groupe car ils sont proches et alignés (loi de similitude de la gestalt) alors qu'ils n'ont de liens entre eux au sein de la structure du menu.

L'alternative retenue est la représentation qui met en valeur les anneaux (Figure 22-a). Elle semble être un bon compromis entre mise en relief de la hiérarchie et maximisation du contexte. En effet, les anneaux renforcent le lien entre les items du même niveau sans masquer totalement le contexte. La hiérarchie est marquée par deux traits fins qui font le lien entre l'item parent et l'anneau fils.

Enfin, les Wave menus permettent la sélection sans regarder (« eyes-free sélection ») afin que l'utilisateur ne soit pas distrait par l'apparition du menu pendant la sélection de commandes. L'utilisateur peut en effet activer des commandes en mode expert et rester concentré sur les objets d'intérêts.



Nous retenons pour les critères d'applicabilité que les Wave menus exploitent au mieux l'espace disponible en reléguant à la périphérie les informations les moins utiles et en conservant au centre, les menus indispensables à l'interaction.

Nous avons présenté les Wave menus relativement à notre espace MenUA. Celui-ci a mis en évidence les avantages théoriques de cette technique pour la navigation en mode novice. Une caractéristique originale des Wave menus est sa représentation concentrique inversée. Pour cette raison, nous avons également développé une variante où la hiérarchie est affichée dans le sens inverse, plus commun, qui est décrite ci-après.

2.4 Wave menus inversés

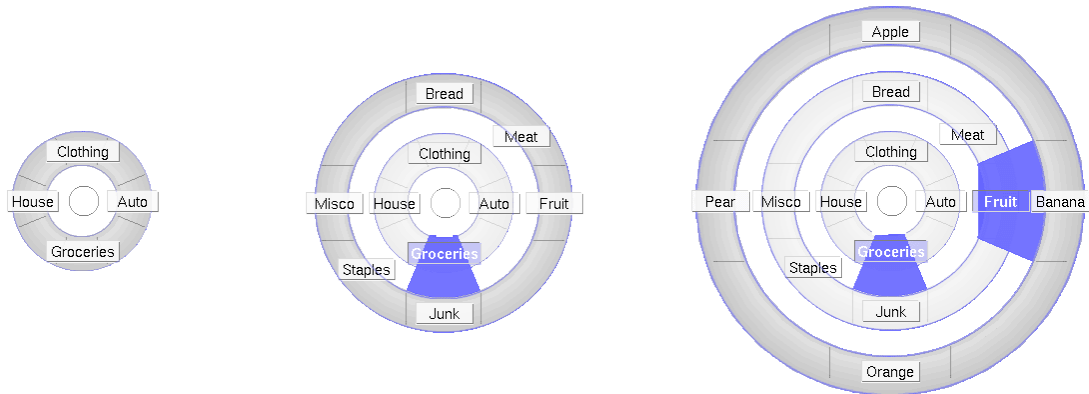


Figure 23 : Wave menus inversés (WM2).

Le Wave menu inversé (WM2) est une technique de menus similaire au Wave menu sauf que les menus sont disposés dans le sens inverse : les menus parents apparaissent au centre et les sous-menus à la périphérie. Ils conservent les trois propriétés interactionnelles qui aident à la navigation (prévisualisation, chemin visible et accès direct aux parents) et ils ont le même mode expert. Cependant, les Wave menus inversés ont certains inconvénients que n'ont pas les Wave menus :

- Les WM2 nécessitent plus d'espace pour être utilisables. En effet, comme les Compound Marking menus, les sous-menus du WM2 peuvent apparaître en dehors de l'écran auquel cas ils ne peuvent plus être activés. Les WM2 ne peuvent donc pas être utilisés sur des dispositifs aux écrans de petite taille.
- Les WM2 peuvent forcer l'utilisateur à faire un détour pour la sélection de certaines commandes Figure 24. Lorsque la commande d'un sous-menu est située dans la direction opposée de l'item parent, un chemin direct sélectionne par erreur un item parent.

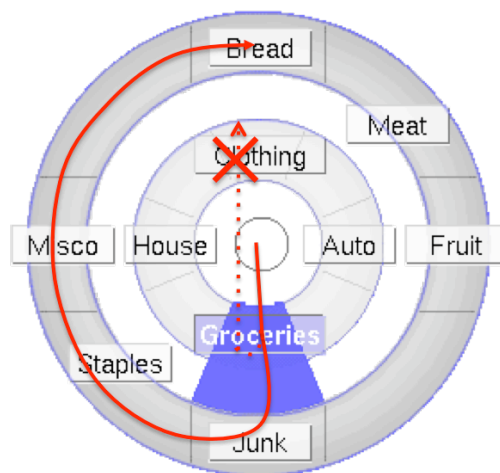


Figure 24 : Détour forcé dans les Wave menus inversés.

Malgré ces inconvénients, il nous semblait intéressant de comparer le Wave menu avec sa représentation duale pour évaluer l'impact de la hiérarchie inversée.

3 Evaluation

Nous décrivons une évaluation comparative du Wave menu avec trois autres menus, le Wave menu inversé, le Compound Marking menu et le Multi-Stroke menu. Notre objectif est de comparer les menus dans le cas des chemins détournés. Nous explicitons d'abord les difficultés de ce d'évaluation puisque les expérimentations existantes concernent principalement le mode expert avec des chemins droits ou directs (section 4 du chapitre 2). Nous décrivons ensuite une tâche paramétrable pour l'évaluation dans le cas de chemins détournés. L'avantage de cette tâche compatible avec les contraintes de l'évaluation contrôlée est de pouvoir fixer le nombre de branches que l'utilisateur doit visiter. Cette tâche est ensuite utilisée pour comparer les performances des quatre menus étudiés.

3.1 Difficultés de l'évaluation pour les chemins détournés

Nous passons en revue les évaluations des menus pour lesquelles les utilisateurs empruntent des chemins directs (évaluation en mode expert) et des chemins droits. Nous exposons ensuite les spécificités et les difficultés de l'évaluation pour des chemins détournés.

Modèle d'évaluation pour des chemins directs (mode expert)

Les Marking menus hiérarchiques ont été évalués pour des utilisateurs expérimentés [Kurtenbach 93, Balakrishnan 98, Zhao 04, Zhao 06]. La difficulté est de pouvoir mettre les utilisateurs dans une situation où ils utilisent le mode expert sans passer par une longue phase d'apprentissage. Pour simuler cette situation, la stratégie consiste à considérer uniquement les gestes du menu (et d'ignorer la représentation du menu). Par exemple, Zhao [Zhao 06] demande aux utilisateurs de reproduire des gestes affichés en haut de l'écran.

Ce type d'expérience a des avantages et inconvénients. Elle requière très peu d'effort cognitif de la part de l'utilisateur [Balakrishnan 98] car le geste à exécuter est fortement corrélé au stimulus. Elle est rapide à réaliser car chaque essai est très court : la reproduction d'une marque dure moins d'une seconde. Enfin, elle permet de s'abstraire complètement de la familiarité de l'utilisateur avec le menu. Les utilisateurs peuvent reproduire les gestes sans connaître le fonctionnement du menu en mode novice.

Le risque de ce type d'évaluation est de généraliser les performances du mode expert à la technique elle-même alors que celui-ci ne représente qu'un cas limite d'utilisation. Par exemple, pour les menus linéaires, seulement quelques commandes sont utilisées en mode expert même après plusieurs années d'utilisation [Lane 05]. De plus cette méthode ne prend pas en compte le temps nécessaire pour que l'utilisateur se rappelle la marque à faire ni les erreurs qu'il risque de commettre. Enfin, ce temps et ce nombre d'erreurs est aussi susceptible de dépendre du nombre de commandes apprises et exécutées en mode expert, un point qui n'est jamais pris en considération par ces études.

Modèle d'évaluation pour des chemins droits (mode novice)

A de rares exceptions, les menus linéaires hiérarchiques sont évalués en mode novice, mais toujours pour des chemins droits [Ahlström 05, Ahlström 06, Cockburn 06, Tsandilas 07, Tanvir 08]. Afin que l'utilisateur suive ce type de chemin pour accéder à une commande, l'item cible est mis en surbrillance ainsi que tous les items qui permettent d'y accéder. Par exemple Ahlström [Ahlström 05] colorie tous les items du chemin à parcourir en vert. L'utilisateur se laisse alors guider par les items colorés sans considérer le nom des items.

Les avantages de ce type d'expérience sont similaires à ceux des expériences sur les techniques en mode expert. Elles requièrent peu d'efforts cognitifs car l'utilisateur se laisse guider. Les essais sont courts : environ 2-3 secondes. Les différences de familiarité entre utilisateurs sont minimales. Ce type d'évaluation reflète le comportement d'utilisateurs familiers,

qui savent exactement comment activer la commande en mode novice, mais pas forcément en mode expert. Cependant, dans la pratique, les utilisateurs utilisent souvent des chemins détournés en particulier pour les commandes moins fréquentes (section 1). La recherche visuelle joue alors un rôle plus important car ils ne savent pas exactement où se trouve la commande recherchée (section 1). Ce type de chemin a été rarement considéré dans les études expérimentales antérieures. Nous exposons maintenant les principales difficultés associées à l'évaluation d'une technique lorsque l'utilisateur doit effectivement naviguer dans le menu.

Evaluation pour des chemins détournés

Il existe une grande variété d'utilisateurs empruntant des chemins détournés selon le niveau d'expertise, de la précision du but et de la stratégie de recherche choisie (section 1). La structure du menu et le vocabulaire utilisés jouent un rôle essentiel dans la manière dont les utilisateurs vont naviguer dans le menu. L'évaluation d'une technique pour des chemins détournés est donc associée à un nombre important de facteurs qu'il est difficile de contrôler précisément dans une évaluation expérimentale. De plus, une différence importante avec les autres types d'évaluation est que les actions de l'utilisateur ne sont pas déterminées à l'avance. L'objectif est ici de définir un stimulus tel que l'utilisateur soit contraint d'emprunter un chemin détourné.

La durée d'un essai est longue car l'utilisateur visite davantage de menus. Or le temps total de l'expérience peut difficilement dépasser 40 minutes car les participants doivent rester concentrés jusqu'à la fin. Ainsi, pour une durée d'expérience comparable à celles des expériences en mode expert (pour lesquelles les essais sont très courts), soit le nombre d'échantillon (d'essai) collecté est réduit soit le nombre de participants doit être augmenté.

Enfin, une tâche de navigation introduit une grande variabilité sur les temps de sélection car la durée d'un essai dépend du nombre de branches visitées. Cette variabilité intra-utilisateurs risque de rendre l'analyse statistique difficile et de produire des résultats non significatifs et non interprétables.

Conscient de ces difficultés, nous définissons dans la section suivante, une tâche pour la comparaison des techniques de menus dans le cas de la navigation.



Nous retenons l'absence d'expérience type pour l'évaluation de techniques de menus pour des chemins détournés.

Les difficultés de ce type d'expériences sont : la durée des essais, la prise en compte de l'expertise des utilisateurs et de la variabilité des résultats.

3.2 Définition et réalisation d'une tâche de navigation

Nous avons vu dans la section 1 deux types de tâches de navigation : le *feuilletage* et la *recherche*. Nous nous intéressons ici à la tâche de recherche car d'une part cette tâche est plus courante. D'autre part, une tâche de feuilletage est caractérisée par l'absence de but précis, ce qui est difficile à modéliser dans une évaluation contrôlée.

Une tâche de recherche avec un chemin détourné consiste à trouver, après avoir visité plusieurs branches, une ou plusieurs cibles. Les cibles peuvent être le nom d'une commande ou simplement une fonctionnalité. La recherche d'une fonctionnalité peut mener à des confusions

par rapport au vocabulaire utilisé. La recherche d'une commande permet de distinguer les erreurs de pointage, des erreurs dues à des confusions.

Un point important est que les utilisateurs naviguent rarement dans la totalité du système de menus mais seulement dans une sous partie. Cette « quantité d'exploration » [Appert 07] dépend de la familiarité de l'utilisateur (section 1). Moins l'utilisateur est familier, plus il visite un nombre important de branches.

Considérer une seule cible risquerait de rendre l'expérience difficilement exploitable à cause d'un effet aléatoire incontrôlé [Appert 07, Bailly 07]. L'utilisateur chanceux trouverait la cible dès la première branche visitée, alors que le moins chanceux trouverait la cible dans la dernière branche visitée. Les temps de sélection pourraient ainsi varier du simple au double selon le nombre de sous-menus visités. Il est préférable de quantifier l'exploration en faisant varier le nombre de branches à explorer (n). Plus le nombre de branches (n) est important, moins l'utilisateur est familier.

Appert [Appert 07] propose une tâche de navigation qui consiste à visiter k distracteurs (branches) avant de pouvoir trouver la cible. Concrètement, celle-ci apparaît dynamiquement dans la k+1^{ème} branche visitée. Cette tâche permet théoriquement d'homogénéiser le nombre de branches visitées. Une expérience préliminaire nous a permis d'observer qu'il était parfois difficile de définir ce que signifie « visiter » une branche. En effet, nos utilisateurs ouvraient parfois un sous-menu accidentellement sans l'explorer. Le seuil pour déterminer quand une branche a été visitée nous semblait difficile à déterminer.

Un autre point important est que l'utilisateur peut manquer la cible dans une tâche de navigation. En effet, il peut balayer les éléments du regard, passer au dessus de la cible et ne pas la voir par manque d'attention ou excès de rapidité (section 1). L'impact sur le temps total de sélection est important car celui-ci peut doubler (nécessité de parcourir de nouveau tous les sous-menus déjà visités). La mise en surbrillance de la cible⁵ évite de la manquer, mais elle introduit un biais qui favorise les techniques qui offrent la prévisualisation. En effet, les items en surbrillance sont plus faciles à détecter avec ces techniques car l'œil est très sensible aux changements de couleurs. Pour ne pas favoriser les techniques avec prévisualisation, il est important que la cible ne se démarque pas artificiellement des autres éléments du système.

Définition de la tâche

La tâche de navigation que nous avons définie permet de contrôler le nombre de branches visitées : l'utilisateur doit *rechercher* toutes les cibles dans un sous-ensemble de l'arbre de commandes. Les cibles sont clairement identifiées grâce à un stimulus textuel.

Pour une hiérarchie à deux niveaux, il y a deux cibles : une pour le premier niveau de la hiérarchie (*cible1*) et une pour le second niveau de la hiérarchie (*cible2*). Plusieurs items *cible1* sont présents au premier niveau. Ils permettent de définir le nombre (n) de sous-menus à visiter. Le nombre de *cible1* détermine donc la quantité d'exploration. Si une seule *cible1* est utilisée (n=1), le visiteur visite un seul sous menu (chemin droit). Plus n augmente, plus le chemin est détourné.

Dans chaque sous-menu de titre *cible1*, il existe plusieurs items *cible2*. Le nombre de *cible2* dans le système de menus est inconnu pour l'utilisateur. Ceci permet d'une part de forcer l'utilisateur à ouvrir tous ces sous-menus et de contrôler le nombre de branches visitées et d'autre part de l'empêcher de savoir s'il a manqué des items cibles. L'utilisateur doit alors sélectionner tous les items *cible2*. Il passe à l'essai suivant lorsqu'il pense avoir sélectionné tous les items *cible2*.

⁵ Les expériences qui évaluent les menus pour des chemins droits mettent l'item cible en surbrillance (fond vert). Les utilisateurs repèrent ainsi très vite l'item cible qui permet de simuler des utilisateurs très familiers.

Chaque sous-menu comporte des items distracteurs. Le nombre d'items distracteurs définit la difficulté du balayage. Plus le nombre d'items distracteurs est élevé, plus le balayage est important et moins les cibles sont mises en évidence.

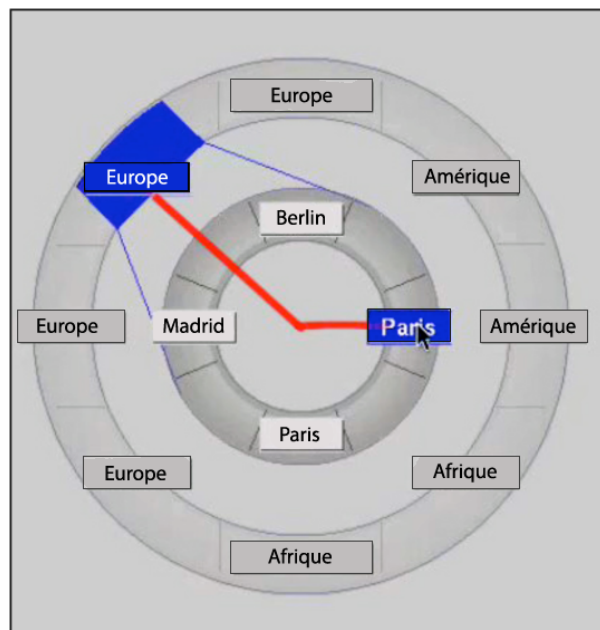


Figure 25 : L'utilisateur doit sélectionner les différents items « Paris » dans les 4 sous-menus « Europe ».

La Figure 25 illustre un exemple avec la commande « Europe » comme *cible1* et « Paris » comme *cible2*. Les utilisateurs doivent alors activer **tous** les items « Europe » (ici, $n=4$) dans le menu racine afin d'explorer les sous-menus correspondants et trouver **tous** les items « Paris ». Le point important est que l'utilisateur doit chercher et sélectionner tous les items « Paris » dans tous les sous-menus « Europe » afin de contrôler la quantité d'exploration. Lorsqu'il pense avoir trouvé tous les items Paris, il passe à l'essai suivant.

Réalisation de la tâche

Pour notre expérience, nous considérons la hiérarchie présentée à la Figure 25 : des noms de continents au premier niveau et des noms de villes appartenant au continent au second niveau. Nous avons utilisé deux niveaux de profondeur afin que chaque essai ne soit ni trop long ni trop lourd cognitivement.

Le nombre d'items *cible1*, c'est-à-dire le nombre de branches à explorer (n), est fixé à 4. Ceci correspond à un utilisateur peu familier mais pas complètement novice. Il aurait été intéressant de faire varier ce paramètre, mais nous avons été limités par la durée totale de l'expérience.

Le nombre d'items *cible2* présents dans la hiérarchie varie entre 2 et 4. Il est balancé entre les techniques afin que la complexité soit la même pour toutes les techniques.

Chaque sous-menu comporte 4, 6 ou 8 items (items *cible2* + items distracteurs) dont la position est contrebalancée afin de ne pas favoriser une direction particulière.

3.3 Expérience

Nous avons utilisé cette tâche pour comparer l'efficacité pour la navigation (rapidité-précision) du Wave menu (WM), du Wave menu inversé (WM2), du Compound Marking menu (CMM) et du Multi-Stroke menus (MSM). Nous les comparons pour des chemins détournés, car pour les chemins directs, le mode de fonctionnement (mode expert) des Wave menus est le même que les Multi-Stroke menus. Pour des chemins directs, les Multi-Stroke menus ont été

prouvé plus efficaces que les Compound Marking menus. Cette expérience complète donc la comparaison des MSM et WM avec les CMM.

Participants

Douze adultes volontaires, 5 femmes et 7 hommes, âgés de 24 à 35 ans (26 en moyenne) ont accepté de participer à notre expérimentation. Ils avaient tous l'habitude d'utiliser un ordinateur et des menus linéaires hiérarchiques. Aucun d'entre eux ne connaissait ni les techniques évaluées ni nos propres techniques.

Matériel

L'expérience a été exécutée sur un ordinateur portable Dell 2GHz pentium 4 avec un écran 17 pouces. Tous les menus ont été réalisés avec la boîte à outils graphique Ubit [Lecolinet 03]. Contrairement aux expériences précédentes avec les Marking menus [Kurtenbach 93, Zhao 04, Zhao 06], nous n'avons pas utilisé de stylet mais une souris Logitech™ comme dispositif d'entrée. La raison principale est que nos utilisateurs n'avaient pas d'expérience avec les tablettes graphiques. Il aurait fallu un temps d'entraînement plus long pour éviter un possible biais. De plus, il nous a semblé important de vérifier la validité des techniques proposées avec les dispositifs matériels les plus couramment utilisés.

Toutes les techniques ont une représentation similaire. Le nom des items est en noir sur fond blanc. Lorsqu'un item est sélectionné, le texte passe en blanc sur fond bleu comme souvent avec les menus linéaires. Le mode expert a été désactivé pour toutes les techniques de même que le délai d'apparition du menu comme nous ne considérons que le mode novice dans cette expérience.

Hypothèses

Les hypothèses sont les suivantes :

H1 : Les Compound Marking menus (CMM) sont plus rapides que les Multi-Stroke menus (MSM) en mode novice car ils possèdent la prévisualisation ainsi que l'accès direct aux parents.

H2 : Les Wave menus (WM) et les CMM ont des performances similaires en mode novice.

H3 : Les Wave Menus (WM) et les Wave Menus inversés (WM2) ont des performances similaires. L'ordre de la hiérarchie n'a pas d'impact sur les performances.

H4 : Il n'y a pas d'effet significatif sur le nombre d'erreurs et le nombre d'items manqués entre les techniques de menus.

Procédure

L'utilisateur teste les 4 techniques de menus à la suite. L'ordre d'apparition des techniques est contrebalancé entre les différents utilisateurs en utilisant un carré latin.

Comme nous souhaitons évaluer la prise en main des différentes techniques, nous laissons l'utilisateur les tester pendant trois minutes sans aucune indication de notre part. Ensuite, nous expliquons le fonctionnement précis de la technique pendant 5 minutes afin d'être sûrs que tout a été compris.

L'utilisateur doit alors exécuter 18 essais. La séquence de hiérarchies a été choisie aléatoirement. Ceci signifie qu'entre chaque essai, la hiérarchie d'items change. Chaque essai marche ainsi : 1) l'utilisateur appuie sur un bouton au centre de l'écran pour commencer l'expérience. 2) Le continent et la ville à rechercher apparaissent en haut de l'écran ; 3) l'utilisateur ouvre le menu en pressant le bouton gauche de la souris ; 4) il peut alors chercher et sélectionner les différentes occurrences du nom de la ville cible dans la hiérarchie ; 5) Il appuie sur la barre d'espace lorsqu'il pense avoir trouvé toutes les occurrences cibles ; 6) Un nouvel

écran apparaît avec un bouton au centre pour commencer un nouvel essai. Les utilisateurs pouvaient prendre une pause entre chaque essai ainsi qu'entre chaque technique.

Résumé

En résumé, l'expérience (les essais de la prise en main exclus) suit la configuration suivante :

12 utilisateurs x
4 techniques de menus (Multi-Stroke, Marking menus Hiérarchique, Wave Menu et Wave menu inversé) x
3 tailles pour les sous-menus (4-6 ou 8 items) x
6 répétitions
= 864 sélections.

Les variables dépendantes sont : le temps de réaction, le temps d'exécution, le temps total, le nombre d'erreurs et le nombre d'items manqués. Le temps de réaction est mesuré à partir de l'apparition du stimulus jusqu'à l'ouverture du menu. Cet intervalle de temps représente le temps nécessaire aux utilisateurs pour comprendre le stimulus avant de faire une sélection. Le temps d'exécution est mesuré à partir de l'ouverture du menu jusqu'à l'appui sur la barre d'espace pour arrêter l'essai. Le temps total est la somme du temps de réaction avec le temps d'exécution. Le nombre d'erreurs est le nombre de sélections incorrectes. Enfin, le nombre d'items manqués est le nombre de villes cibles qui n'ont pas été sélectionnées par les utilisateurs alors qu'elles étaient présentes dans le menu.

3.4 Résultats

Une analyse de la variance à deux facteurs n'indique pas d'effet significatif sur le temps de réaction pour les techniques de menus.

Temps d'exécution

Le temps d'exécution est calculé à partir des essais sans item manqué. L'analyse de la variance révèle un effet important sur le temps d'exécution pour les *techniques de menus* ($F_{3,33} = 47.25$, $p < .0001$). Un test post hoc Tukey avec un alpha à 1% montre que le WM et le CMM sont significativement plus rapides que le MSM et le WM2. En moyenne, les WM réduisent le temps de sélection de 18% (respectivement 17% pour les CMM) par rapport aux MSM.

Comme attendu, nos résultats révèlent aussi un effet pour la *taille des sous-menus* ($F_{2,22} = 68.69$, $p < .0001$). Le temps de sélection augmente avec le nombre d'éléments. Ceci s'explique par le fait que la phase de recherche visuelle est évidemment plus longue lorsque le nombre d'éléments à lire est plus important.

On observe également un léger effet pour l'interaction *taille des sous-menus*technique* ($F_{6,66} = 2.75$, $p < 0.01$) ce qui indique que le nombre d'éléments distracteurs par sous-menus affecte les techniques différemment. Une comparaison par paires (test Tukey à 1%) n'indique pas de différence entre 6 et 8 items pour le MSM. Il montre également que le WM2 est significativement plus lent pour 6 et 8 items. Une explication vraisemblable est que la distance entre les items des sous-menus est plus importante dans le cas des WM2. Le balayage de tous les éléments nécessite des sauts visuels qui ralentissent la lecture. Il serait intéressant de le vérifier avec une analyse oculométrique.

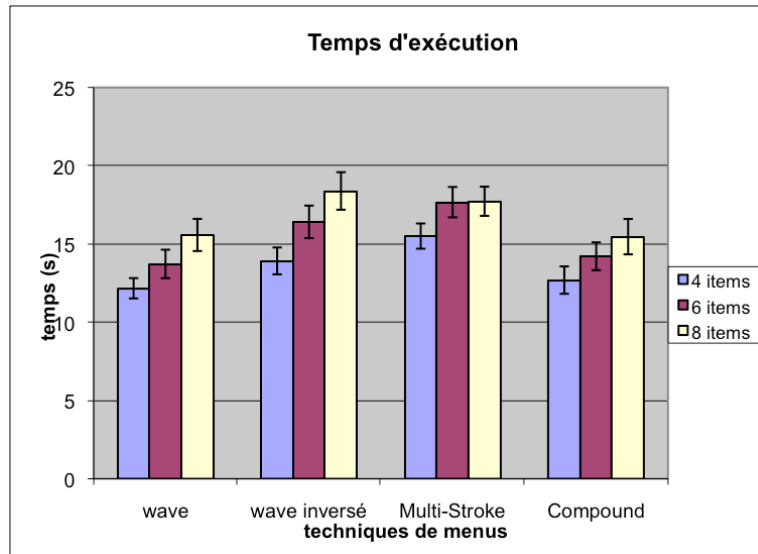


Figure 26 : Temps d'exécution en secondes en fonction des techniques et du nombre d'items dans les sous-menus.

L'analyse du temps total montre les mêmes effets significatifs que pour le temps d'exécution (car il n'y a pas d'effet sur le temps de réaction).

Taux d'erreurs

Nous constatons un effet significatif du nombre d'erreurs sur les techniques ($F_{3,33} = 13.59, p < .0001$). Un test Tukey à 1% montre que le nombre de sélections incorrectes est significativement plus élevé pour le MSM qui produit plus de 3 fois plus d'erreurs que les autres techniques. Ceci peut s'expliquer par le fait que les utilisateurs ont des difficultés pour sélectionner le cercle central qui permet de remonter dans la hiérarchie avec le MSM. En voulant rapidement revenir en arrière, les utilisateurs activent un item par erreur. Finalement, Il n'y a pas d'effet significatif sur le nombre d'éléments par sous-menus.

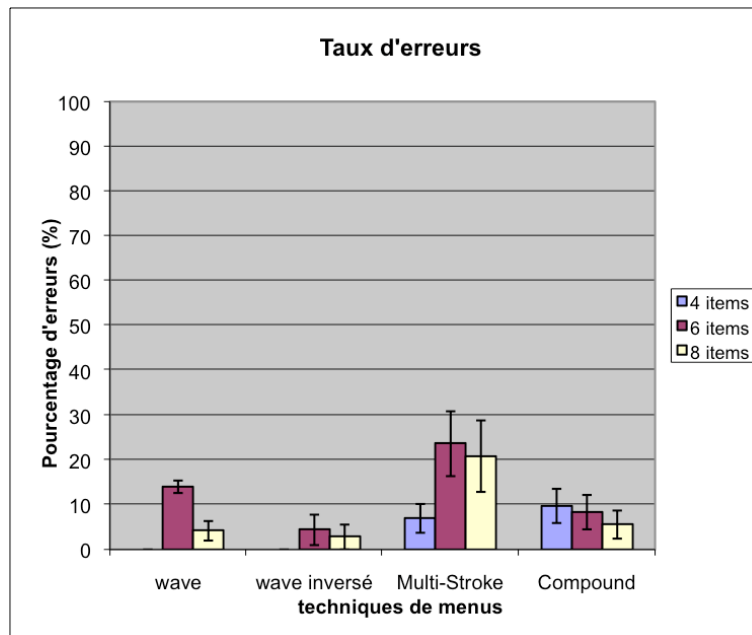


Figure 27 : Taux d'erreur en fonction des techniques et du nombre d'items dans le menu.

Taux d'items manqués

L'analyse de la variance ne révèle pas d'effet significatif sur le nombre d'items manqués.

Evaluation subjective

Après l'expérience, nous avons demandé aux utilisateurs de classer les quatre menus par ordre croissant pour les critères : préférence, prise en main, fatigue, perception de la hiérarchie

6 utilisateurs sur 12 ont classé le CMM comme leur technique préférée et 5 ont choisi le WM. 9 utilisateurs sur 12 ont classé le MSM en dernière position.

5 utilisateurs ont jugé que le CMM était la technique la plus facile à apprendre, suivi des WM2 (3) et des WM (2). A l'opposé, 6 utilisateurs ont considéré le MSM comme la technique la plus difficile à apprendre et 4 les WM.

7 utilisateurs ont jugé que le MSM était la technique la plus fatigante, 2 le CMM et le WM2. A l'opposé, 7 utilisateurs ont trouvé que le CMM est la technique la moins fatigante, 2 ont choisi le WM et le MSM.

8 utilisateurs ont trouvé plus difficile de percevoir la hiérarchie avec le MSM et 3 avec le WM2. A l'opposé, la hiérarchie a été mieux perçue avec le CMM pour 7 utilisateurs. 2 utilisateurs ont choisi le WM et le WM2.

Enfin, 7 utilisateurs ont déclaré ne pas avoir été gênés par la hiérarchie inversée utilisée dans le WM. 3 ont été gênés seulement au début et 2 durant la totalité de la session.

3.5 Discussions

Nous pouvons maintenant revenir sur nos hypothèses au regard des résultats de l'expérimentation.

Les CMM et les WM sont plus rapides que les MSM

Notre expérience montre que les CMM sont 18% plus rapides que les MSM pour notre tâche de navigation. Les deux propriétés, prévisualisation et accès direct aux parents, semblent avoir un impact important sur la navigation dans ces techniques de menus. 11 utilisateurs sur 12 ont déclaré qu'il était facile de se perdre dans le MSM car ils avaient du mal à se rappeler s'ils avaient déjà visité une branche. L'hypothèse 1 est donc vérifiée.

Les WM et les CMM partagent des performances similaires en mode novice ce qui renforce l'hypothèse selon laquelle c'est bien les deux propriétés de prévisualisation et d'accès direct aux parents qui sont importantes. L'hypothèse 2 est également vérifiée.

L'ordre de la hiérarchie (WM, WM2) a de l'importance

Nos résultats montrent que la technique des Wave menus est en fait plus rapide que la variante inversée. Ceci peut être expliqué par le fait que les items balayés par l'utilisateur dans le WM2 sont plus éloignés que dans le Wave menu. La somme des distances entre les items pour chacune de ces techniques est équivalente car elles partagent la même représentation concentrique. Cependant, pendant l'interaction, le regard ne parcourt pas les items de la même manière. Une caractéristique importante des WM est que le menu le plus profond de la branche visible est toujours au centre de la représentation. La distance entre les items de ce sous-menu est faible. L'utilisateur porte donc principalement son attention sur le sous-menu le plus compacté. A l'opposé le menu le plus profond est à l'extérieur avec les WM2. Les items de ce sous-menu sont plus éloignés et l'utilisateur est forcé de faire des sauts visuels importants. Les utilisateurs ont confirmé ce point en indiquant que la distance entre les items à lire avec les WM2 était plus importante. L'hypothèse 3 n'est donc pas vérifiée.

Les MSM ont un taux d'erreur plus important que les autres techniques

Nous avons observé que le Multi-Stroke menu est significativement moins précis. Il produit plus de 3 fois plus d'erreurs que les autres techniques. En observant les utilisateurs, nous avons noté qu'ils avaient des difficultés à cliquer dans la zone centrale pour revenir au menu parent. Ils sélectionnaient souvent un item par erreur. Ce problème ne peut pas se produire en mode expert car l'utilisateur ne revient jamais en arrière.

Nous n'avons pas trouvé de différence significative entre les techniques pour le nombre d'items manqués. L'hypothèse 4 est donc vérifiée que pour les items manqués.

Prise en main des Wave menus

L'évaluation subjective montre que les utilisateurs avaient des difficultés à prendre en main les Wave menus et ont jugé les CMM plus facile à apprendre. Ces difficultés découlent sûrement de la représentation non-conventionnelle des Wave menus. Aussi, nous avons proposé une nouvelle représentation des Wave menus qui ne modifie pas l'interaction. Cette représentation, illustrée Figure 28, invite l'utilisateur à élargir le menu courant pour laisser apparaître progressivement le sous-menu. Elle est basée sur la métaphore de la pile où tous les sous-menus seraient superposés.



Figure 28 : Wavelet menu : amélioration du Wave menu pour l'aspect apprentissage de la technique. La représentation est basée sur une métaphore de pile.



Nous retenons que les Wave menus et les Compound Marking menus sont plus rapides et plus précis que les Multi-Stroke menus pour une tâche de navigation.

Les utilisateurs accueillent agréablement les Wave menus et l'idée d'une représentation de la hiérarchie inversée.

4 Conclusion

Nous nous sommes intéressés dans ce chapitre à l'exploration dans les systèmes de menus. Nous avons souligné l'importance du mode novice par le fait qu'une grande variété d'utilisateurs sont amenés à naviguer dans les systèmes de menus et à emprunter des chemins détournés. Nous avons également explicité les propriétés interactionnelles impliquées dans la navigation : la prévisualisation, l'observabilité du chemin parcouru et l'accès direct aux items parents.

Nous avons pris les Marking menus hiérarchiques comme cas d'étude. Alors que la plupart des travaux portent sur les performances en mode expert, nous avons choisi d'améliorer le mode novice.

Nous avons proposé une nouvelle technique de menus appelée les Wave menus que nous avons décrite au regard de notre espace MenUA. L'originalité de cette technique est de présenter la hiérarchie en anneaux concentriques inversés. Les sous-menus sont affichés au centre, alors que les menus parents sont relégués à l'extérieur. Cette représentation spatiale conserve les 3 propriétés utiles pour la navigation dans la hiérarchie de commandes.

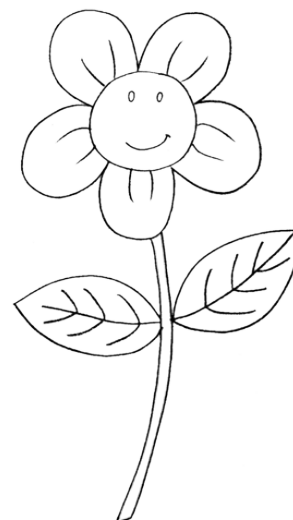
Cette technique a l'avantage d'exploiter au mieux l'espace disponible. Les Wave menus peuvent être utilisés en mode dégradé sur de petites surfaces d'entrée et de sortie (contrairement aux Compound Marking menus), mais lorsque la surface d'affichage est suffisante, la technique permet de naviguer efficacement dans la hiérarchie (contrairement aux Multi-Stroke menus).

Une autre contribution réside dans la définition d'une tâche de navigation pour l'évaluation. Nous avons défini une tâche qui force l'utilisateur à explorer une partie de la hiérarchie de telle sorte que les résultats soient statistiquement exploitables. Nous avons réalisé une expérience qui repose sur cette tâche. Elle montre que les Multi-Stroke menus sont moins efficaces que les Compound Marking menus pour des chemins détournés. Ces résultats vont à l'encontre de ceux obtenus pour des chemins directs ce qui souligne l'importance de l'évaluation pour différents types de chemins. De plus, l'expérimentation montre que les Wave menus sont 18% plus rapide que les Multi-Stroke menus.

Les Wave menus améliorent donc les Marking menus à la fois pour les critères d'utilisabilité et pour les critères d'applicabilité de l'espace MenUA. En mode novice, ils ont les avantages des Compound Marking menus en permettant de naviguer efficacement dans la hiérarchie de commandes sans les inconvénients, car ils sont utilisables sur de petites surfaces d'entrée/sortie. En mode expert, ils partagent les avantages des Multi-Stroke menus grâce à l'utilisation de marques simples qui garantissent un haut niveau de précision.

**De la discrimination à l'apprentissage des gestes :
Les Flower menus, une solution pour les Marking menus**

Chapitre 7



1	ORGANISATION DES MENUS	150
1.1	BALANCEMENT LARGEUR VS. PROFONDEUR	150
1.1.1	GROUPEMENTS	150
1.1.2	RAPIDITE	150
1.1.3	PRECISION	151
1.1.4	CHARGE COGNITIVE	151
1.1.5	APPRENTISSAGE	152
1.1.6	ESPACE ECRAN	152
1.2	FORME DES MENUS	153
1.3	APPLICATIONS EXISTANTES	154
1.4	CONCLUSION : LE CAS DES MARKING MENUS	156
2	GESTES DES MARKING MENUS	156
2.1	NOMBRE MAGIQUE 7	156
2.2	GESTES DES MARKING MENUS	157
2.3	INTRODUCTION DE LA POSITION RELATIVE (ZONE ET POLYGON MENUS)	157
2.4	INTRODUCTION DE LA COURBURE ET DU SENS	158
3	EVALUATION DES GESTES COURBES	161
3.1	EXPERIMENTATION	161
3.2	RESULTATS	163
3.3	SYSTEME DE RECONNAISSANCE GESTUEL	164
4	FLOWER MENUS	165
4.1	LIMITATIONS DES MARKING MENUS	166
4.2	PRINCIPE DES FLOWER MENUS	167
4.3	PROPRIETES DES FLOWER MENUS	169
4.3.1	RAPIDITE ET PRECISION	169
4.3.2	APPRENTISSAGE ET MEMORISATION	171
4.3.3	SATISFACTION	172
4.3.4	ADEQUATION A L'APPLICATION	172
4.3.5	ADEQUATION A LA PLATEFORME	173
4.3.6	ADEQUATION A LA TACHE	173
5	EVALUATION	176
5.1	EXPERIENCES EXISTANTES	176
5.2	EXPERIMENTATION	178
5.3	RESULTATS	181
5.4	DISCUSSIONS	183
5.5	APPRENTISSAGE ET MEMORISATION : PISTES DE REFLEXION	185
6	CONCLUSION	188

Bien que l'efficacité des Marking menus a été prouvée depuis longtemps [Kurtenbach 91], leur utilisation reste marginale dans les interfaces graphiques. Une possible explication est le nombre d'items qu'ils peuvent contenir en largeur (à un même niveau de profondeur). Les Marking menus sont en effet limités à 8 commandes du fait d'un vocabulaire gestuel basé uniquement sur l'orientation. Pour combler cette lacune, nous proposons d'élargir leur vocabulaire gestuel en exploitant de nouvelles dimensions comme la courbure.

Un avantage majeur des Marking menus est qu'ils favorisent l'apprentissage implicite du mode expert. L'argument principalement utilisé est la répétition : les utilisateurs exécutent le même geste en modes novice et expert. Nous mettons en évidence que la répétition n'est vraisemblablement pas le seul facteur qui rentre en jeu. Plus généralement, ce chapitre concerne l'impact du vocabulaire gestuel sur la capacité des utilisateurs à apprendre rapidement le mode expert. Il détaille donc à la fois le critère d'utilisabilité *apprentissage et mémorisation* et le critère d'applicabilité *adéquation à l'application* que nous avons déjà caractérisé dans MenUA au chapitre 4, section 2 et au chapitre 5, section 3.

La section 1 est consacrée à l'organisation des commandes dans les menus. Nous soulignons le besoin d'avoir des structures hiérarchiques pouvant contenir un grand nombre de commandes (en particulier en largeur) et mettons en évidence les limites des Marking menus pour ce type d'organisation.

La section 2 est dédiée aux gestes utilisés dans les Marking menus. Nous rappelons les limites d'un vocabulaire gestuel basé uniquement sur l'orientation. Nous proposons d'étendre ce vocabulaire en exploitant également la courbure et le sens afin de contenir un grand nombre de commandes en largeur.

La section 3 présente une expérimentation visant à évaluer la capacité des utilisateurs à dessiner correctement ces marques courbées.

La section 4 propose une nouvelle technique de menus exploitant ces marques courbées, intitulée *Flower menus*. Les Flower menus augmentent le nombre de commandes des Marking menus sans dégrader l'apprentissage du mode expert. Les propriétés interactionnelles des Flower menus sont décrites au regard de notre espace MenUA.

La section 5 présente une évaluation expérimentale visant à comparer l'impact des Flower menus et de deux autres techniques de menus sur la rapidité de l'apprentissage du mode expert. Nous mettons en évidence l'influence d'autres facteurs que la répétition pour ce critère et proposons quelques explications possibles.

1 Organisation des menus

L'organisation des menus est un problème essentiel pour les concepteurs d'applications. Quelle doit être la profondeur de la hiérarchie ? Combien de commandes doivent figurer à chaque niveau ? Comment grouper les commandes ? Toutes ces questions ont été abordées depuis longtemps [Snoberly 83, Kieger 84] et pourtant il n'existe toujours pas de consensus. De plus, avec le développement des sites marchands sur la toile qui sont forcés d'organiser leurs produits dans des menus, ces questions sont toujours d'actualité [Zaphiris 02, Bernard 02] : la structure adoptée a un impact significatif sur les ventes.

La structure d'un menu se caractérise principalement par sa *largeur* (nombre maximum de commandes à un niveau) et sa *profondeur* (nombre maximum de niveaux dans la hiérarchie). Le problème posé aux concepteurs d'application est d'optimiser la largeur et la profondeur des différents sous-menus afin de faciliter l'interaction et la mémorisation (de cette hiérarchie). Nous discutons les différentes alternatives puis nous les confrontons à nos observations de 6 logiciels grand public. Le constat est que le nombre de commandes à un niveau est généralement supérieur au nombre maximal de commandes que peut contenir un Marking menu.

1.1 Balancement Largeur vs. Profondeur

Nous étudions les avantages et les inconvénients du « balancement » largeur vs. profondeur au regard de 7 critères : groupements, rapidité, précision, charge cognitive, apprentissage et espace écran.

1.1.1 Groupements

Les structures hiérarchiques nécessitent de trouver des noms aux sous-menus. Ce travail est souvent délicat pour les concepteurs [Snyder 85] car le nom doit à la fois : 1) refléter les commandes du menu, 2) ne pas évoquer les commandes des autres sous-menus, 3) être court et 4) être compréhensible par l'utilisateur [Norman 91]. Plus la structure est profonde, plus les concepteurs doivent trouver de titres. Les titres en haut de la hiérarchie se révèlent souvent être plus abstraits car ils englobent davantage de catégories qui sont elles-mêmes déjà abstraites. Il devient difficile d'avoir des catégories qui ne se recouvrent pas partiellement en haut de la hiérarchie. Ces recouvrements ont un impact important sur la surcharge cognitive de l'utilisateur (section 1.1.4), la navigation (section 1.1.2) et l'apprentissage (section 1.1.5). Les structures profondes dégradent donc la qualité des groupements.

Les groupements internes (évoqués à la section 1 du chapitre 4) constituent une alternative aux groupements hiérarchiques. Ce sont des groupements sémantiques à un même niveau de profondeur séparés par une ligne et qui ne nécessitent pas de nom. Ils permettent de subdiviser une longue liste en groupes logiques qui facilitent la recherche d'une fonctionnalité.

Les groupements internes permettent donc de structurer les commandes sans augmenter la profondeur. Ils ont les avantages des menus hiérarchiques comme favoriser la recherche visuelle sans les inconvénients (augmentation de la profondeur, difficulté de trouver un nom, etc.)

1.1.2 Rapidité

Recherche visuelle

Les utilisateurs qui ne connaissent pas le nom de la fonctionnalité recherchée sont forcés de lire la plupart des libellés des items pour faire leur choix (inspection en série, section 1.4 du chapitre 6). Une organisation en profondeur permet de réduire le nombre d'items à lire car elle permet aux utilisateurs d'éliminer plusieurs branches à chaque niveau.

Au contraire, les structures en largeur favorisent la recherche visuelle des utilisateurs plus expérimentés. Ces derniers peuvent balayer efficacement une longue liste de commandes, voir dans certains cas, poser leur regard directement sur le bon item (section 1.4 du chapitre 6). Les utilisateurs éliminent d'eux-mêmes les informations non pertinentes et les séparateurs des groupements internes favorisent la recherche visuelle en offrant des points de repères visuels.

Sélection d'un item

La sélection d'un item est généralement plus rapide avec une organisation en largeur car la longueur du geste à effectuer dépend de la profondeur du menu [Zhao 04]. Comme pour l'accès à une page dans les sites web, une organisation en largeur réduit le nombre de clics nécessaire [Larson 98, Zaphiris 02].

Landauer et Nachbar [Landauer 85] ont analysé le balancement largeur vs. profondeur pour la sélection dans des menus linéaires sur des écrans tactiles en utilisant un modèle simplifié qui repose sur la loi de Fitts [Fitts 54] et la loi de Hick-Hyman [Hick 52, Hyman 53]. Ils concluent qu'une organisation en largeur donne généralement une meilleure performance qu'une organisation en profondeur. Landauer et Nachbar ont effectué une expérience avec 4096 items arrangés dans des sous-menus contenant 2, 4, 8 ou 16 éléments pour des utilisateurs empruntant un chemin droit. Leurs résultats montrent que la largeur maximale (16 items) améliore les performances globales. Nous notons deux limitations à cette étude. D'une part, elle s'intéresse uniquement aux menus linéaires et d'autre part elle considère seulement les chemins droits alors que la structure du menu peut influencer le nombre de branches visitées et le temps total pour atteindre une cible.

Navigation

Les organisations en profondeur nécessitent souvent davantage de navigation. Nous avons vu précédemment, que le titre des sous-menus dans les structures profondes tend à être plus abstrait (section 1.1.1). Dans ce contexte, l'utilisateur peut prendre une décision erronée et parcourir une mauvaise branche. Il doit alors naviguer plus longtemps pour trouver la commande recherchée.

Snoberry [Snoberry 83] montre que le nombre de branches visitées par erreur augmente de 4% à 34% lorsque la profondeur du menu augmente de 1 à 6 niveaux. Les organisations en profondeur ont tendance à augmenter la quantité d'exploration dans les systèmes de menus.

Nous abordons maintenant le balancement largeur vs. profondeur pour la précision. Ce critère est critique pour les menus circulaires.

1.1.3 Précision

Le balancement largeur vs. profondeur peut être lié à la capacité des utilisateurs à sélectionner une cible précisément. Nous rappelons que dans le cas des menus circulaires, augmenter la largeur du menu conduit à réduire l'espace angulaire alloué à chaque item. Ces derniers sont alors plus difficiles à sélectionner. Kurtenbach [Kurtenbach 93a] montre pour les Compound Marking menus qu'il est difficile d'avoir plus de 8 items en largeur et plus de 2 niveaux de profondeur en mode expert. Les Multi-Stroke menus [Zhao 04] ou les Wave menus (présenté dans le chapitre 6) augmentent la profondeur du menu sans dégrader la précision. Nous proposons à la section 4 une nouvelle technique de menus permettant d'accueillir un nombre élevé d'items en largeur.

1.1.4 Charge cognitive

Un utilisateur novice peut être surchargé cognitivement face à un nombre important d'items présentés simultanément (structure en largeur). Il doit lire, mais surtout, mémoriser plusieurs items avant de faire son choix (section 1.4 du chapitre 6). Lorsque ce nombre est trop

élevé, la mémorisation est difficile et l'utilisateur doit relire une partie des items. Les ergonomes recommandent de ne pas dépasser 7 ± 2 items à chaque niveau de la hiérarchie ce qui est le nombre maximum d'items que l'utilisateur peut enregistrer dans sa mémoire à court terme [Miller 56].

Les groupements internes et les groupements hiérarchiques guident l'utilisateur pas et pas et permettent de réduire la charge cognitive. Néanmoins, les structures profondes ont davantage de catégories qui se recouvrent, car elles sont plus abstraites. L'utilisateur est contraint de fournir un effort supplémentaire pour choisir les branches à visiter.

1.1.5 Apprentissage

Un utilisateur est d'autant plus encouragé à explorer un système d'information (menus ou site web) qu'il est capable de se représenter correctement la hiérarchie. La question est de savoir « quels types de structures aident l'utilisateur à se construire rapidement une représentation mentale de la hiérarchie » [humanf. 04]. Il est plus difficile de construire une représentation mentale d'une hiérarchie organisée en profondeur [humanf. 04].

1.1.6 Espace écran

L'espace écran disponible pour afficher des commandes est limité, en particulier sur les dispositifs mobiles. Ces dernières ne peuvent pas afficher simultanément plus de 10 items verticalement et 3 horizontalement. Il est alors nécessaire de créer des groupes hiérarchiques. De plus, menu organisé en largeur masque davantage le contexte de l'utilisateur.

	Avantage d'une structure en largeur	Avantage d'une structure en profondeur
Recherche visuelle	Balayage des items plus facile	Réduction du nombre d'items à lire
Sélection d'un item	Chemin plus court	
navigation	Réduction du nombre de branches visitées par erreur	
précision		Geste plus précis pour les menus circulaires
Charge cognitive	Noms des sous-menus moins abstraits	Réduction du nombre d'alternatives
Apprentissage	Construction plus facile d'une représentation mentale de la hiérarchie	
Espace écran		Moins d'items affichés simultanément

Tableau 1 : Synthèse du balancement largeur vs. profondeur.

Les principaux résultats sur le balancement largeur vs. profondeur sont résumés dans le tableau 1. La plupart des études concluent qu'il est préférable d'avoir une organisation en largeur plutôt qu'en profondeur que ce soit pour les menus d'applications [Snoberry 83, Kieger 84, Landauer 85, Jacko 96] ou pour les menus des sites web [Larson 98, Zaphiris 02]. L'organisation optimale dépend d'un grand nombre de facteurs comme la qualité des catégories, l'ordre des éléments, la taille de l'écran, les techniques d'interaction disponibles. Ces résultats sur le balancement « largeur vs. profondeur » ne s'appliquent pas aux cas des menus audios qui n'utilisent pas la modalité visuelle [Zhao 08] car ceux-ci ne permettent pas de balayer les items

d'un coup d'œil et sont fortement contraints par la mémoire de travail de l'utilisateur [Marics 97, Balentine 99, Cohen 04, Schumacher 95]. La largeur optimale des menus audios est autour de 4-5 items [Marics 97, Balentine 99, Cohen 04, Schumacher 95] et peut atteindre 11 lorsque les items sont familiers [Commarford 08]. Enfin, Norman [Norman 88] s'est intéressé à un aspect complémentaire : la forme de la structure des menus (Figure 1).

1.2 Forme des menus

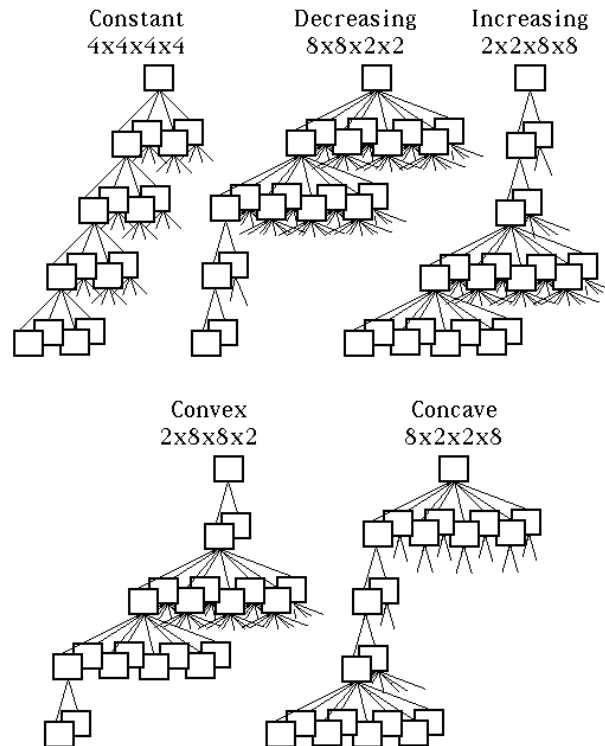
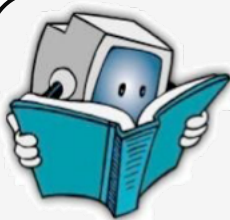


Figure 1: Différentes structures de menus. Pour le même nombre de commandes (256) et la même profondeur (4), Norman [Norman 88] étudie la variation de la largeur en fonction de la profondeur. Illustration extraite de [Norman 91].

La Figure 1 illustre différentes formes (concave, convexe, etc.) de menus pour le même nombre de commandes (256) et la même profondeur (4). Dans cette étude, Norman [Norman 88] fait varier la largeur des menus en fonction de la profondeur. Il observe que lorsque la cible est clairement définie, la structure « Increasing » (Figure 1), c'est-à-dire la structure qui augmente la largeur des menus avec la profondeur est légèrement meilleure. Cependant, lorsque la cible est moins bien définie, la structure concave (Figure 1), c'est à dire la structure avec des menus moins larges aux niveaux intermédiaires est bien meilleure. Plus récemment, Bernard [Bernard 02] montre également que les structures concaves favorisent la navigation. Ces résultats sont en accord avec la section 1.3 du chapitre 6 où nous indiquons que les utilisateurs étaient moins familiers avec les niveaux intermédiaires de la hiérarchie quelque soient leurs connaissances et leurs expertises.



Nous retenons que pour les menus visuels, les organisations en largeur sont généralement préférables aux organisations en profondeur quelque soit le type d'utilisateurs.

Les groupements internes sont très utiles car ils jouent un rôle sémantique pour les utilisateurs inexpérimentés et fournissent des points de repères visuels pour les utilisateurs expérimentés.

Nous proposons maintenant d'étudier l'organisation de quelques applications bien connues afin de vérifier si elles adoptent des structures en largeur ou en profondeur.

1.3 Applications existantes

Nous avons réalisé une analyse de la structure des menus de 6 applications grand public du système d'exploitation Windows. Les applications considérées sont : Office Word 2003, Office Excel 2003, Adobe Reader 7, Mozilla Firefox 2.0, Mozilla Thunderbird 0.9 et Photoshop 7.0. Nous avons choisi ces applications car elles sont largement répandues et concernent des domaines d'applications variés. Nous avons étudié la taille des menus, le nombre de groupements internes et leur taille. L'étude porte sur les menus linéaires attachés à la barre de menus¹.

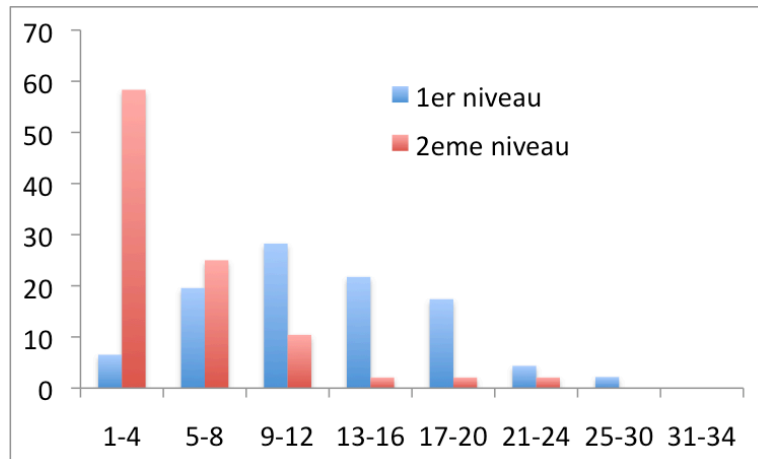


Figure 2 : Distribution du nombre d'items (en %) dans les menus (1^{er} niveau) et les sous-menus (2^{ème} niveau). Par exemple, la configuration la plus répandue consiste en des sous-menus avec 1-4 commandes.

Nous observons que les menus contiennent souvent plus du maximum de 9 (7+2) commandes recommandées par les ergonomes. La moyenne est de 12.4 commandes au premier niveau (Beaudouin-Lafon avait trouvé une moyenne de 12.5 dans [Beaudouin-Lafon 97]) et de 5.5 au second niveau (3.9 dans [Beaudouin-Lafon 97]). La Figure 2 montre la distribution des commandes dans les menus et les sous-menus. Nous observons en particulier que 21.7% des menus contiennent entre 13 et 16 commandes et 23.9% des menus contiennent plus de 16 commandes. La Figure 4 montre le menu « Affichage » du Logiciel Microsoft Word contenant 24 commandes. La profondeur des menus est rarement supérieure à 2 niveaux de hiérarchie, (3 en réalité car le premier niveau est affiché dans la barre de menus). Les autres commandes sont alors accessibles par des boîtes de dialogues. Les applications courantes adoptent donc des organisations en largeur ce qui est conforme avec l'analyse faite à la section précédente.

De plus, pour les groupements internes, nous avons observé que tous les menus du premier niveau contenaient des groupements internes, utiles pour réorganiser les menus larges. Les menus contiennent en moyenne 4.9 groupements internes au premier niveau². La Figure 3 montre la distribution des commandes dans les groupements internes. On observe que 58.2% des groupes contiennent moins de deux items et 91.9% des groupes contiennent moins de 4 items. A part Photoshop qui contient de longues listes de filtres et d'effets, tous les groupements internes ont moins de 7 items. Les groupements internes sont donc très fréquents et de petite taille.

¹ Il aurait été intéressant d'élargir cette étude aux menus contextuels mais il est difficile d'obtenir la liste de tous les menus contextuels différents d'une application

² Les sous-menus contiennent en moyenne 1.9 groupements. Cette diminution s'explique par le fait qu'un nombre important de sous-menus contienne uniquement 1 ou 2 commandes.

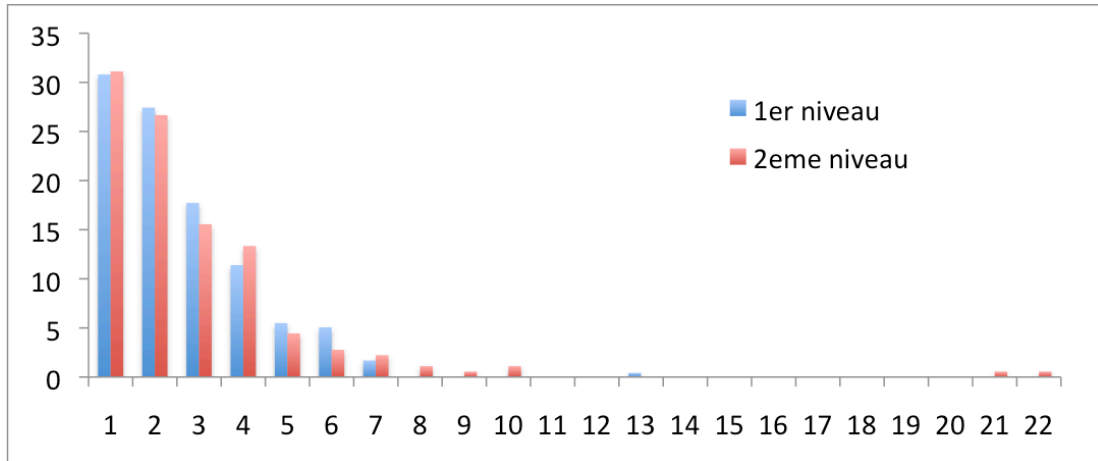


Figure 3 : Distribution du nombre d'items dans les groupements internes des menus (1^{er} niveau) et sous-menus (2^{ème} niveau).

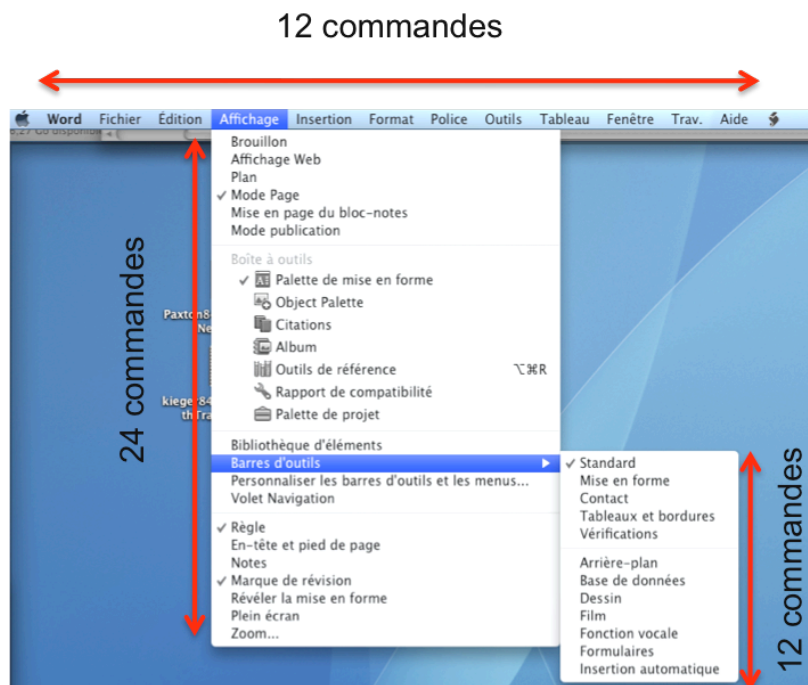


Figure 4 : Largeur de la barre de menus et de deux sous-menus dans Microsoft Word.



Nous retenons que les applications courantes ont des menus qui contiennent souvent plus de 12 commandes et peuvent facilement avoir jusqu'à 16 commandes. Les groupements internes sont très fréquemment utilisés et contiennent habituellement moins de 7 items.

1.4 Conclusion : le cas des Marking menus

Les Marking menus sont limités à 8 commandes en largeur [Kurtenbach 93a] (section 1.1.3). Ils ne sont donc pas compatibles avec les organisations en largeur des applications existantes car leur contenance est nettement inférieure à la moyenne observée (12.4), sans mentionner les menus ayant un très grand nombre de commandes (comme par exemple le sous-menu « Affichage » de Microsoft Word). Dans ce contexte, la limitation en largeur implique d'adopter une structure hiérarchique pour utiliser les Marking menus. La structure résultante peut être artificielle et donc ne pas refléter la représentation mentale de l'utilisateur (section 2.3 du chapitre 4). Une solution pour pallier à ce problème est d'accroître le nombre de gestes reconnus.

2 Gestes des Marking menus

Cadoz [Cadoz 94] distingue :

- les gestes épistémiques qui permettent d'explorer le monde ;
- les gestes sémiotiques qui permettent de communiquer ;
- les gestes ergotiques qui permettent d'agir matériellement sur le monde.

Les gestes considérés dans les Marking menus sont des gestes ergotiques. Ils permettent d'activer des commandes qui vont agir sur l'interface. Contrairement aux interfaces standard, où seules la position initiale et la position finale sont interprétées, l'*interaction gestuelle* exploite la dynamique du geste. L'interaction gestuelle a plusieurs avantages :

- elle transmet davantage d'informations au système (tous les points entre le point de départ et le point d'arrivée peuvent être exploités) ;
- elle exploite notre capacité musculaire à mémoriser et reproduire des trajectoires (capacité que nous employons notamment pour écrire) ;
- elle s'abstrait du positionnement absolu et permet ainsi la sélection sans regarder.

Cette section souligne les limites du vocabulaire gestuel des Marking qui repose uniquement sur l'orientation. Pour cela, nous nous appuyons d'abord sur les travaux de « jugements absolus » de Miller [Miller 56]. Nous exposons ensuite un vocabulaire alternatif qui considère en plus de l'orientation, la position relative d'une marque par rapport à une origine. Ce vocabulaire, exploité par les Zone et Polygon menus [Zhao 06] permet d'associer plus de commandes en largeur mais comporte certaines limitations. Nous proposons alors un vocabulaire gestuel qui exploite l'orientation, la courbure et le sens.

2.1 Nombre magique 7

En 1956, Miller [Miller 56] propose une règle générale sur la limite de la mémoire instantanée autour de 7 éléments (appelés « mnèmes »). Un mnème constitue la plus petite unité de mémoire (chapitre 4). Un mnème peut être une tonalité, une orientation, mais aussi, un chiffre, une lettre, un mot ou même une phrase. Par exemple, lorsque des participants doivent se rappeler une série de mots non corrélés, la tâche est difficile à partir de 6 ou 7 mots (alors que chaque mot contient plusieurs lettres). De même, lorsque des participants doivent distinguer entre plusieurs tonalités, si le nombre de tonalités présentées est au dessus de 5, la précision de cette tâche chute rapidement. En d'autres termes, la mémoire instantanée impose de sévères limitations sur le nombre d'éléments que l'on est capable de percevoir, traiter et se rappeler. C'est le cas en particulier pour l'orientation, la seule dimension utilisée par les Marking menus classiques (section 2.2).

Miller ajoute également que lorsque le stimulus varie selon plusieurs dimensions (variation en X, Y au lieu de simplement en X), on peut améliorer la précision des jugements. La **combinaison** de plusieurs dimensions augmente ainsi la limite de la mémoire instantanée.

Miller annonce alors que la limite se trouverait au alentour d'une centaine d'éléments. Nous verrons, dans la section 2.4, que nous avons adopté cette idée pour définir un nouveau vocabulaire gestuel en combinant orientation, courbure et sens.

2.2 Gestes des Marking menus

Kurtenbach [Kurtenbach 93a] montre que l'on peut difficilement exploiter plus de 8 ou 12 orientations dans les Marking menus si l'on souhaite conserver un niveau de précision élevé en mode expert. Miller [Miller 1956] rapporte que l'on est capable de discriminer entre 6.9 orientations (correspondant à 2.8 bits d'information) pour une exposition d'1/40 de seconde et 9.8 (3.3 bits) pour une exposition de 5 secondes.

Pour augmenter le nombre de commandes, il est donc nécessaire d'élargir le vocabulaire gestuel et de considérer de nouvelles dimensions.

2.3 Introduction de la position relative (Zone et Polygon menus)

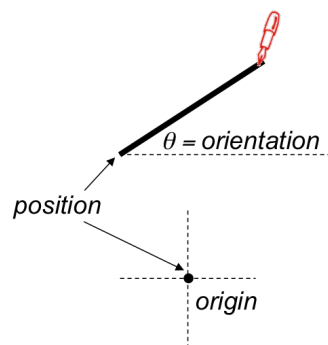


Figure 5 : Zhao [Zhao 06] considère en plus de l'orientation, la position relative par rapport à une origine (le point d'activation du menu). Illustration extraite de [Zhao 06].

Nous rappelons que les Zone et Polygon menus sont deux types de Marking menus, illustrés Figure 15 et Figure 16, qui prennent en compte en plus de l'orientation, la position relative des marques par rapport à l'origine du geste (Figure 5). Le geste de l'utilisateur est donc en deux étapes : il spécifie d'abord l'origine par un clic (qui est le point d'activation du menu), puis se décale pour faire une marque avec une certaine orientation (θ). Dans [Zhao 06], 8 orientations et quatre zones différentes sont considérées. Le vocabulaire contient donc 32 gestes qui sont illustrés Figure 6. Ces gestes peuvent être exécutés très précisément car seulement 8 orientations sont utilisées [Zhao 06].

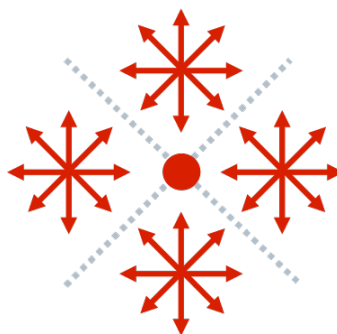


Figure 6 : Tous les gestes utilisés (32) dans les Zone menus [Zhao 06].

L'un des inconvénients des Zones menus est l'occupation spatiale. Zhao propose un nouveau menu, les Polygon menus (Figure 16), qui retiennent les 16 gestes de la Figure 7 mais qui les organisent différemment afin d'avoir une représentation plus compacte (Figure 16) que les zones menus.

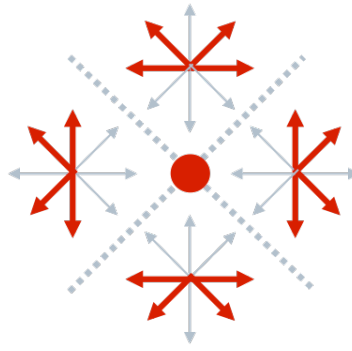


Figure 7 : Les 16 gestes retenus pour les Polygon menus [Zhao 06].

En résumé, ces deux ensembles de gestes permettent d'augmenter le nombre de commandes en largeur des Marking menus. Cependant, ces gestes ne sont plus radiaux par rapport au point d'origine des Marking menus originels. De plus, le geste de sélection d'un item implique deux actions : un tap puis une marque. Des travaux récents proposent d'exploiter en plus de l'orientation, la distance [Nancel 08], impliquant ainsi qu'une seule action plutôt que deux (section 1.1 du chapitre 5).



Nous retenons que les Zone et Polygon menus augmentent le vocabulaire de gestes en considérant en plus de l'orientation, la position relative de la marque par rapport à une origine (le point d'activation du menu).

Nous proposons un autre vocabulaire de gestes ne nécessitant qu'une seule action.

2.4 Introduction de la courbure et du sens

L'originalité de notre vocabulaire de gestes est de considérer en plus de l'orientation, deux nouvelles dimensions : *la courbure* et *le sens*. Ces gestes sont utilisés dans une nouvelle technique de menus appelée Flower menus qui est présentée dans la section suivante. Nous décrivons maintenant chacune de ces deux dimensions.

Gestes							
nom	P-	C-	B-	S	B+	C+	P+
Temps d'exécution (ms)	937	817	708	498	704	813	931

Figure 8 : Les 7 gestes du Flower menu pour l'orientation nord et leur temps d'exécution respectifs.

Courbure du geste

En géométrie, la courbure en un point d'une courbe représente le cercle qui approxime au mieux cette partie de la courbe. Les gestes droits représentent seulement un cas particulier de courbure dont le degré est nul. Nous considérons 4 degrés de courbure différents, illustrés Figure 8 pour l'orientation nord:

- **S** : les gestes droits (Straight). Ce sont les gestes de base des Marking menus.
- **B** : les gestes inclinés (Bent). Ils sont légèrement infléchis par rapport aux gestes droits ($\approx 40^\circ$).
- **C** : les gestes de rebroussements (Cusp). Ils se distinguent des gestes inclinés par le fait qu'ils reviennent en arrière ($\approx 130^\circ$).
- **P** : les gestes boucles (Pigtail). Ils sont caractérisés par le fait que la courbure est importante ($> 180^\circ$). Il n'est pas nécessaire qu'il y ait un point d'intersection (ou quasi-boucle).

Nous nous sommes limités à 4 degrés de courbure afin que l'utilisateur puisse facilement les discriminer. Miller [Miller 56] montre que la courbure est l'une des dimensions les plus difficile à "juger en absolu" (« absolute judgement »). Les utilisateurs peuvent difficilement discriminer plus de 3 degrés de courbure (1.6 bits) lorsque la longueur de la corde est constante et 4 degrés de courbure (2.2 bits) lorsque la longueur de l'arc est constante (Figure 9). Comme, nous n'imposons pas de contraintes sur la longueur de la corde, 4 degrés de courbure semble être un optimum. En utilisant plus de courbures, on augmenterait significativement le taux d'erreurs. En utilisant moins de courbures, on se priverait d'une ressource utile en entrée.

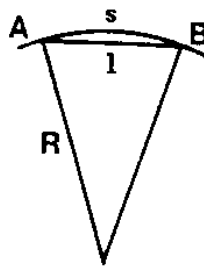


Figure 9 : L'arc (S) et la corde (I) de courbe entre les points A et B et le rayon R.

Sens du geste

La seconde dimension est le sens. Elle indique de quel côté est la courbure :

- - : sens anti-horaire. La courbure se dirige dans le sens inverse des aiguilles d'une montre (sens trigonométrique) ;
- + : horaire. La courbure se dirige dans le sens des aiguilles d'une montre.

Le sens permet de doubler le nombre de gestes courbés (sauf pour les gestes droits qui ont un degré de courbure nul).

Vocabulaire de 56 gestes

Ces deux dimensions peuvent être combinées avec l'orientation. Ainsi, pour chaque orientation, nous disposons de 7 gestes. Ces gestes sont illustrés Figure 8 pour l'orientation nord. En considérant les 8 orientations, nous disposons d'un vocabulaire de $8 \times 7 = 56$ gestes différents illustrés Figure 10. Ce design correspond bien aux critères énoncés dans Miller (section 2.1) dans la mesure où nous combinons 3 dimensions comportant chacune un nombre proche ou inférieur au nombre magique (8 orientations, 4 courbures, 2 sens). Les formes résultantes devraient donc être plus facile à discriminer (percevoir et traiter) et mémoriser que si nous n'utilisions qu'une dimension unique comportant un nombre équivalent de valeurs.

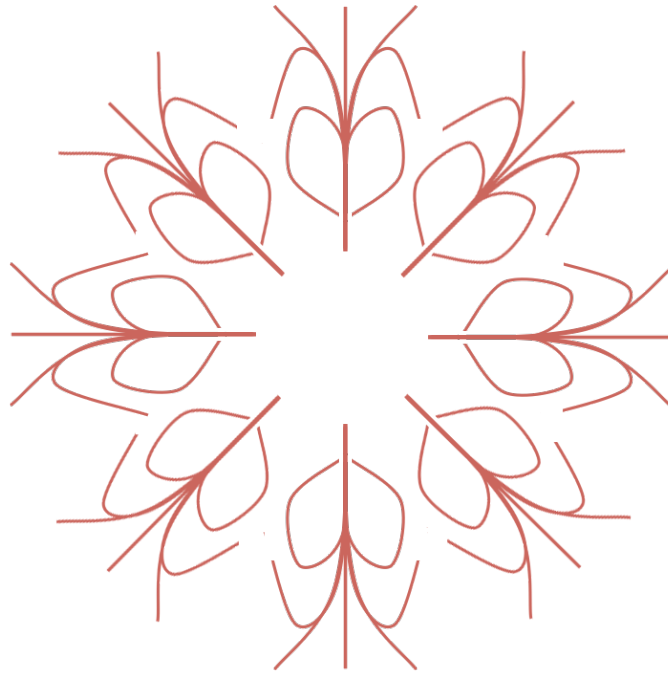
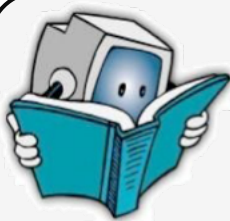


Figure 10 : Les 56 gestes des Flower menus. $7 = 2 * 4 - 1$ gestes pour chacune des 8 orientations.

Les gestes courbes ont été utilisés par d'autres techniques d'interaction. Les FlowMenus [Guimbretiere 00] utilisent des boucles pour la sélection et l'activation d'une commande. Hinckey [Hinckley 05] utilise la fermeture d'une boucle (pigtail) pour délimiter la saisie de texte de la sélection de commandes. Enfin, les Hybrid Marking menus [Isokoski 02], conçus pour l'entrée de texte, augmentent les Marking menus de gestes courbés sur les axes principaux (horizontaux et verticaux). Nous nous différencions de ces travaux par le fait que nous considérons ces gestes de manière systématique et pour toutes les orientations. Nous élargissons la taille du vocabulaire d'entrée et nous combinons tous ces gestes dans une seule technique : les Flower menus (Figure 18).



Nous retenons qu'en combinant l'orientation avec la courbure et le sens, nous disposons d'un vocabulaire de 56 gestes. Ces gestes sont utilisés dans les Flower menus.

Avant d'introduire ces gestes dans une technique de menus, il est nécessaire de vérifier la capacité des utilisateurs à les exécuter et les différencier. Nous décrivons dans la section ci-après une expérimentation des tracés de ces gestes.

3 Evaluation des gestes courbés

Nous avons étudié expérimentalement la manière dont les utilisateurs réalisent les gestes courbés (performance moteur). Les objectifs de cette expérience étaient :

- de vérifier la capacité des utilisateurs à dessiner tous ces gestes de manière suffisamment précise ;
- d'obtenir des données expérimentales afin d'avoir une base d'apprentissage et une base de test pour développer un système de reconnaissance efficace ;
- d'identifier les gestes les plus efficaces (en rapidité-précision) afin de guider les concepteurs pour le positionnement des commandes fréquentes.

Les gestes considérés sont les gestes droits (S), inclinés (B), de rebroussements (C) et les boucles (P) pour les deux sens et les 8 orientations (Figure 10).

La démarche et la procédure de cette expérimentation sont semblables à celles de Moye [Moyle 02] et Buisson [Buisson 03] sauf qu'elles considéraient uniquement les gestes droits dans les Marking menus classiques.

3.1 Expérimentation

Hypothèses

Les hypothèses que nous voulons vérifier sont :

H1 : *Le temps d'exécution augmente avec la degré de courbure.* Cet effet est prédit par la loi de la puissance 2/3 [Viviani 82] qui affirme que $A(t) = K C(t)^{2/3}$ avec A l'accélération angulaire, C la courbure et K un paramètre : un facteur de gain d'accélération.

H2 : *Il existe une erreur angulaire systématique pour toutes les orientations.* Moye [Moyle 02] observe que les utilisateurs dessinent les lignes droites avec un angle légèrement différent de l'angle théorique.

H3 : La variabilité angulaire est plus grande sur les axes diagonaux. [Zhao 04] observe plus d'erreurs sur des axes diagonaux pour les Marking menus classiques. Ceci peut être dû à une variabilité plus importante des marques sur ces axes.

Tâche et stimulus

Pour vérifier nos hypothèses, la tâche consiste à dessiner aussi vite et aussi précisément que possible 56 gestes (8 directions * 7 gestes). Le stimulus est une représentation graphique de la marque à reproduire. Cette représentation, utilisée également dans [Balakrishnan 98, Zhao 06], a été choisie pour minimiser l'effort cognitif car elle permet à l'utilisateur de comprendre facilement le geste à dessiner.

Procédure

Avant l'expérience, les participants sont informés que nous nous intéressons aux propriétés des gestes courbés. Nous insistons sur le fait que la taille des marques n'a aucune importance et qu'ils doivent prendre des pauses pour ne pas se fatiguer le poignet.

Chaque essai est organisé ainsi : a) le participant appuie sur le bouton « Start » ; b) le stimulus apparaît (la marque à reproduire) en haut de l'écran et un cercle apparaît au centre de la zone de dessin ; c) l'utilisateur dessine la marque en commençant dans le cercle et en maintenant le bouton de la souris enfoncé ; d) l'essai est terminée lorsque l'utilisateur relâche la souris. La marque disparaît alors de l'écran ; e) un nouveau stimulus apparaît et ainsi de suite. Aucun retour visuel autre que la marque produite n'est donné aux utilisateurs.

Au cours d'un bloc (ensemble d'essais), les utilisateurs dessinent les 64 (8 orientations * 4 degrés de courbure * 2 sens) gestes à la suite, une fois pour chaque geste, sauf pour les gestes droits qui sont dessinés deux fois (ceux-ci étant identiques dans les deux sens). Nous avons cependant besoin de deux échantillons pour les gestes droits afin d'avoir le même nombre d'éléments pour chaque degré de courbure. Pour illustrer l'expérience, la Figure 12 montre l'ensemble des gestes inclinés dans le sens horaire pour tous les participants.

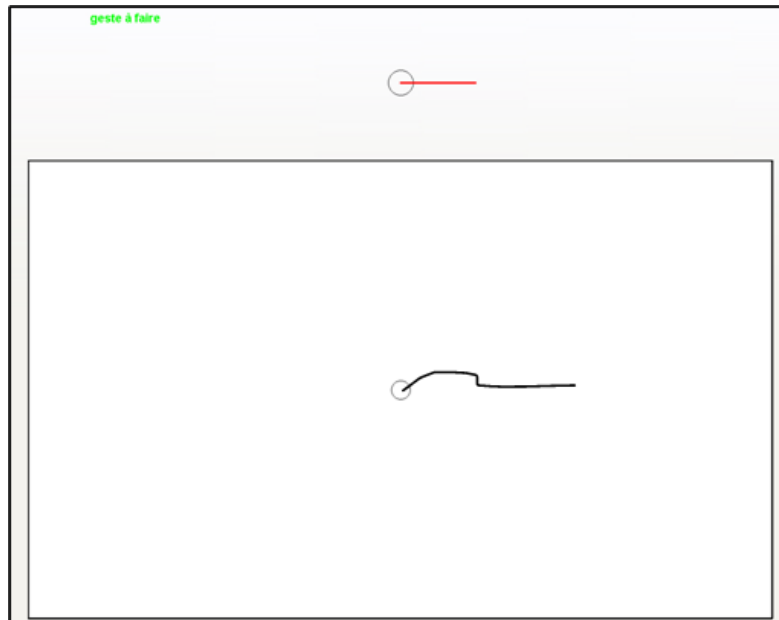


Figure 11 : L'interface graphique de l'expérience. Le stimulus est représenté en haut de l'écran. Le cercle indique le début du geste. Similairement, un cercle au centre de l'écran indique à l'utilisateur l'endroit où il doit commencer son geste.

Participants

14 participants (5 femmes, 9 hommes) ont été recrutés au sein de l'Université. Ils ont entre 21 et 30 ans (moyenne 25). Ils sont tous droitiers et familiers avec l'informatique. Ils ont reçu une poignée de bonbons pour leur participation.

Appareils

L'expérience a été conduite sur un ordinateur portable Dell Latitude D800, 2GHz pentium 4 avec un écran 17 pouces. L'interface a été réalisée en C++/Qt. Le dispositif de pointage est une souris Logitech 3 boutons. Comme pour l'expérience des Wave menus (section 3 du chapitre 6), nous avons choisi d'utiliser une souris plutôt qu'un stylet pour 2 raisons : la souris reste le dispositif le plus utilisé et des études montre que le stylet donnent des résultats supérieurs ou égaux à la souris [Kurtenbach 93a, Kurtenbach 93b]. Ainsi, si ces gestes sont efficaces dans le cas le plus défavorable, ils le seront aussi en utilisant le stylet.

Design

Nous avons utilisé un plan d'expérience intra-participants. Les trois variables, orientation, courbure et sens sont changées après chaque essai. Une séquence d'orientations a été fixée aléatoirement permettant d'alterner axes principaux et axes diagonaux. Chaque participant commençait l'expérience à une position différente (décalée d'une unité) dans la séquence. La présentation de la courbure et du sens était aussi contre-balançée pour chaque orientation.

Chaque geste est dessiné 5 fois : une fois dans chaque bloc (à l'exception des gestes droits qui étaient dessinés deux fois par bloc, mais pas de manière consécutive). Chaque

participant réalise l'expérience en une session qui dure environ 20 minutes. Aucun participant n'a indiqué avoir rencontré de la fatigue au niveau du poignet.

En résumé, le design de l'expérience est le suivant :

14 participants *
 8 orientations *
 4 courbures *
 2 sens *
 5 répétitions
 = 4480 marques

3.2 Résultats

Nous avons réalisé une 8*4*2 (Orientation, courbure, sens) intra-participant analyse de la variance.

Orientation	Execution mean (ms)	Execution sd	Angular error mean (°)	Angular error (sd)
East	720	353	-0.8	6.6
North-East	732	379	-2	10.7
North	739	387	0.6	5.8
North-West	753	376	3.1	13
West	717	327	0.3	6
South-West	759	365	1.6	11.4
South	746	357	1.4	6.5
South-East	743	360	3.4	10.4

Tableau 2 : Le temps d'exécution et l'erreur angulaire pour chaque orientation.

Temps d'exécution

Comme prévu par la loi de la puissance 2/3 [Viviani 82], nos résultats montrent que le temps d'exécution augmente avec le degré de courbure ($F_{3,39} = 62.09$, $p < 0.0001$). Un test de Tukey montre que les gestes droits (498ms) sont plus rapides que les gestes inclinés (704ms) qui sont plus rapides que les rebroussements (813ms) qui sont eux même plus rapides que les boucles (929ms). L'hypothèse H1 est donc vérifiée et les commandes les plus fréquentes doivent être associées aux gestes les moins courbés.

Erreur angulaire

Les données expérimentales montrent un biais entre l'angle initial (mesuré entre le point de départ et un point à 60% de la longueur totale) et l'angle du stimulus. Ceci est probablement dû au fait que les utilisateurs utilisaient la souris sur le côté droit de l'ordinateur. Ceci suggère également que l'effet sera différent pour des utilisateurs gauchers ou pour des utilisateurs utilisant une tablette PC. Un système de reconnaissance de gestes efficace devrait pouvoir personnaliser ces effets aux gauchers ou à des dispositifs particuliers. L'hypothèse H2 est également vérifiée.

Variabilité

L'analyse de la variance montre un effet significatif de l'orientation sur la variabilité des angles initiaux ($F_{7,91} = 31.69$, $p < 0.0001$) que l'on peut voir sur la Figure 12. Un test Tukey montre que la variabilité est plus importante sur les axes diagonaux que sur les axes principaux. Il sera par conséquent préférable d'utiliser d'abord les gestes disponibles sur les axes

principaux. Cependant, cet effet devrait pouvoir être facilement compensé par un système de reconnaissance de geste efficace (section 3.3).

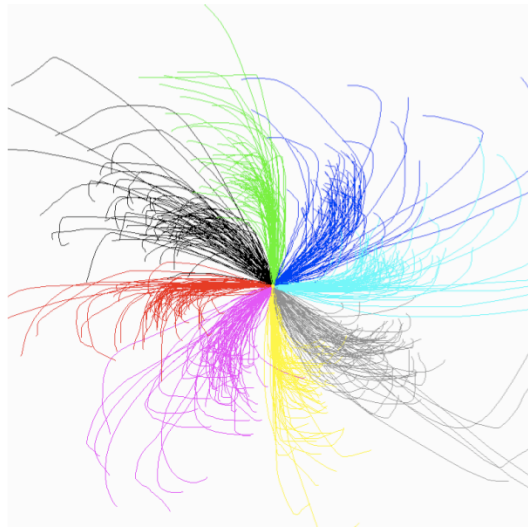


Figure 12 : Ensemble des gestes inclinés dans le sens anti-horaire pour tous les utilisateurs.

Ces résultats peuvent être appliqués aux Marking menus classiques pour améliorer leurs performances en tenant compte de l'erreur angulaire et de la variabilité sur les gestes. Même si visuellement, les secteurs apparaissent homogènes, ils peuvent être différents en entrée comme le montre la Figure 13.

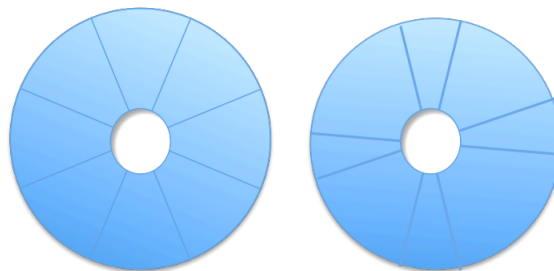


Figure 13 : A gauche, un Marking menu équilibré. A droite, un Marking menu prenant en compte les biais de l'utilisateur (secteur plus large sur les diagonales et centré sur les axes préférentiels).

3.3 Système de reconnaissance gestuel

Nous avons développé un système de reconnaissance des gestes qui prend précisément en compte la manière dont les utilisateurs dessinent les marques. En effet, celui-ci ne doit pas interpréter comme erronés des gestes qui sont en réalité corrects. Notre système de reconnaissance est basé sur l'algorithme des « K plus proche voisins » qui comporte l'avantage d'être à la fois rapide et précis dans le cas concerné. Cet algorithme, très simple, est basé sur une approche à base de votes. Une distance est associée à chaque élément de la base pour un nouveau geste inconnu. Parmi les K gestes qui minimisent cette distance, un vote est ensuite effectué pour déterminer le geste élu. Dans notre cas, nous avons utilisé la distance euclidienne et $K=5$. Nos primitives étaient simplement l'angle de départ sur $2/3$ de la longueur et l'angle d'arrivée sur le $1/3$ restant. Ces primitives ont été utilisées pour calculer la distance. Nos tests montrent qu'ils sont suffisants pour différencier tous les gestes. ce système pourrait être amélioré en considérant des primitives plus évoluées.

Les échantillons de la moitié des utilisateurs ont été utilisés comme base d'apprentissage et l'autre moitié pour les tests. Nous avons supprimé de la base, les gestes erronés (erreurs de consigne) ou illisibles (environ 2% de tous les gestes). Pour ce faire, un humain devait reconnaître chaque geste. S'il le reconnaissait différemment du geste théorique, celui-ci était

supprimé de la base. En effet, un geste considéré faux par un lecteur humain ne doit pas servir d'exemple pour le système de reconnaissance.

Ce système est suffisamment rapide pour fournir un retour visuel immédiat (à chaque nouvelle position du dispositif d'entrée). Le taux de reconnaissance est de 99% pour les 24 premières commandes (gestes droits et inclinés) ; 96.5% si l'on considère les 40 premières commandes (en ajoutant les rebroussements) et 93% pour toutes les commandes réunies. Il faut de plus noter que tous ces gestes ne seront pas utilisés simultanément : par exemple, pour une technique de menus, une vingtaine de gestes courbes suffisent. Dans ce cas, une grande partie des gestes de rebroussements et les boucles ne seront pas utilisés et le taux de reconnaissance sera certainement au-dessus de 96.5%.

Les gestes courbes peuvent donc à la fois être dessinés précisément par les utilisateurs et interprétés correctement par le système. C'est un pré-requis indispensable pour les utiliser efficacement dans une technique de menus.



Nous retenons que les gestes courbés peuvent à la fois être dessinés précisément par l'utilisateur et interprétés correctement par le système.

4 Flower menus

Cette section décrit une nouvelle technique de menus appelée Flower menus qui augmente le nombre de commandes en largeur sans pour autant dégrader l'apprentissage du mode expert. Pour cette raison, les Flower menus utilisent les gestes courbés présentés et évalués dans les sections précédentes. Nous exposons d'abord le principe de fonctionnement des Flower menus puis nous en détaillons les propriétés interactionnelles au regard de notre espace MenUA. Préalablement, nous rappelons les limitations des différents types de Marking menus qui nous ont amenées à concevoir les Flower menus.

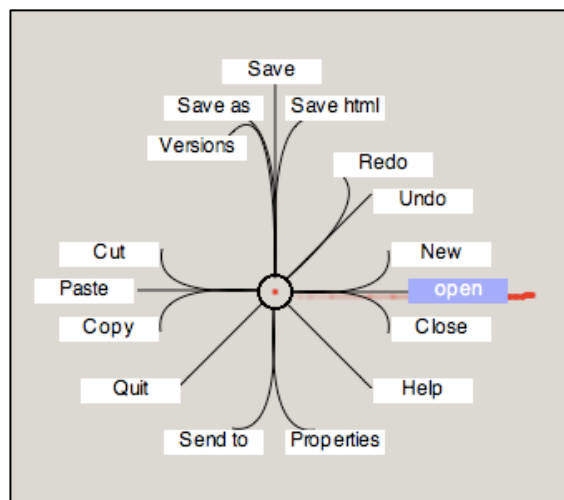


Figure 14 : les Flower menus

4.1 Limitations des Marking menus

Les Compound Marking menus [Kurtenbach 93a], les Multi-Stroke menus [Zhao 04] et les Wave menus [Bailly 07] sont limités à 8-12 commandes à chaque niveau de profondeur car ils considèrent uniquement l'orientation. Les Zone et les Polygon menus prennent en compte en plus de l'orientation des marques, la position relative de ces marques par rapport au point d'activation du menu (section 2.3).

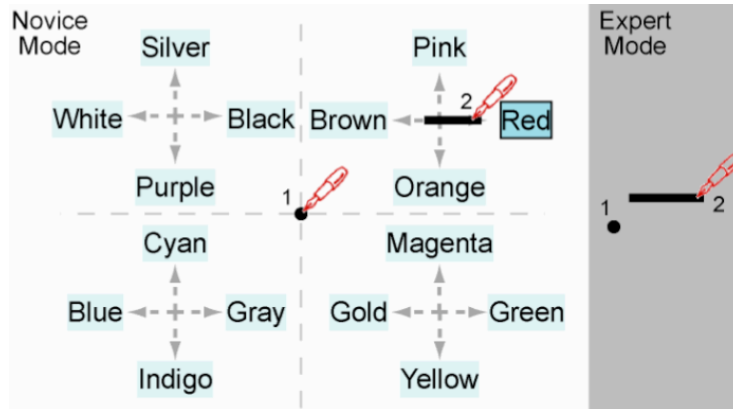


Figure 15 : Zone Menus (ZM) [Zhao 06]. Illustration détaillée dans la section 1 du chapitre 5.

Les Zone menus (ZM) (Figure 15) peuvent contenir jusqu'à 32 commandes. Cependant, ils nécessitent beaucoup trop d'espace écran pour être utilisables en tant que menus contextuels (4 fois plus que les menus linéaires). Ils dégradent donc le critère *adéquation à la plateforme* (dispositifs de sortie) pour améliorer celui d'*adéquation à l'application* (Nombre d'items en largeur) de l'espace MenUA.

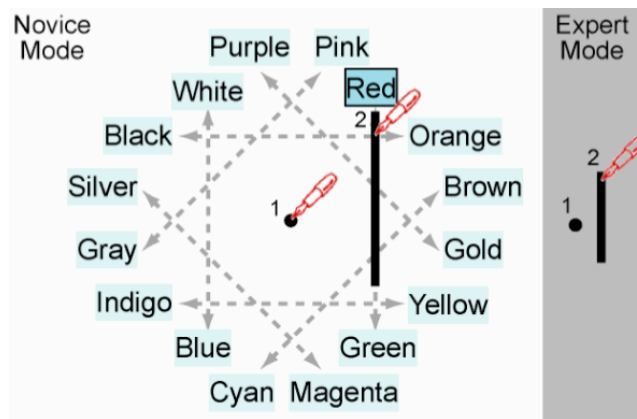


Figure 16 : Les Polygon menus (PM) [Zhao 06]. Illustration détaillée dans la section 1 du chapitre 5.

Les Polygon menus (Figure 16) utilisent le même principe que les Zone Menus. Ils offrent une représentation plus compacte et peuvent contenir 16 commandes. Cependant, nous estimons qu'en améliorant le critère *adéquation à l'application* (nombre d'items en largeur), ils dégradent celui d'*apprentissage et mémorisation* (mode expert) de l'espace MenUA. Nous verrons lors d'une expérience contrôlée (section 5) que les utilisateurs ont des difficultés à apprendre les gestes associés aux commandes à cause de leur présentation particulière.

Finalement, face au problème de l'occupation spatiale des Marking menus, Tapia [Tapia 95] a adopté une représentation compacte qui rend difficile les groupements internes (Figure 17). Ces derniers sont pourtant forts utiles (section 1.1) et largement répandus dans les menus linéaires.

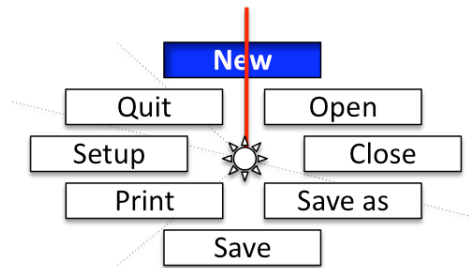


Figure 17 : Représentation compactée des Marking menus qui ne permet plus d'utiliser des séparateurs (en pontillés) pour les groupements internes.



Nous retenons qu'il n'existe pas de solutions pleinement satisfaisantes au problème du nombre de commandes en largeur des Marking menus.

Nous retenons aussi l'absence des groupements internes dans les Marking menus malgré leur utilité et leur omniprésence dans les menus linéaires.

4.2 Principe des Flower menus

Les Flower menus sont des Marking menus qui reposent sur le vocabulaire gestuel présenté à la section 2.4 et évalué à la section 3. Ils considèrent en plus de l'orientation, la courbure et le sens. Les différents types de gestes sont illustrés en situation à la Figure 18. Les Flower menus généralisent les Marking menus classiques qui ne considèrent que les gestes droits. Ils ne nécessitent qu'une seule opération, le dessin de la marque, contrairement aux Zone et Polygon menus.

Les Flower menus fonctionnent comme les Marking menus. Ils ont deux modes de fonctionnement : le mode novice et le mode expert.

Lorsque l'utilisateur presse le bouton de la souris et attend 100ms, le menu entre en mode novice (Figure 19-a) et apparaît centré autour du curseur. Chaque commande est associée à une marque qui commence au centre du menu. L'ensemble de ces marques courbées suggère les pétales d'une fleur. L'utilisateur sélectionne une commande en reproduisant simplement la marque associée. Il peut à tout moment annuler la sélection en revenant dans la zone centrale et en relâchant le bouton de la souris.

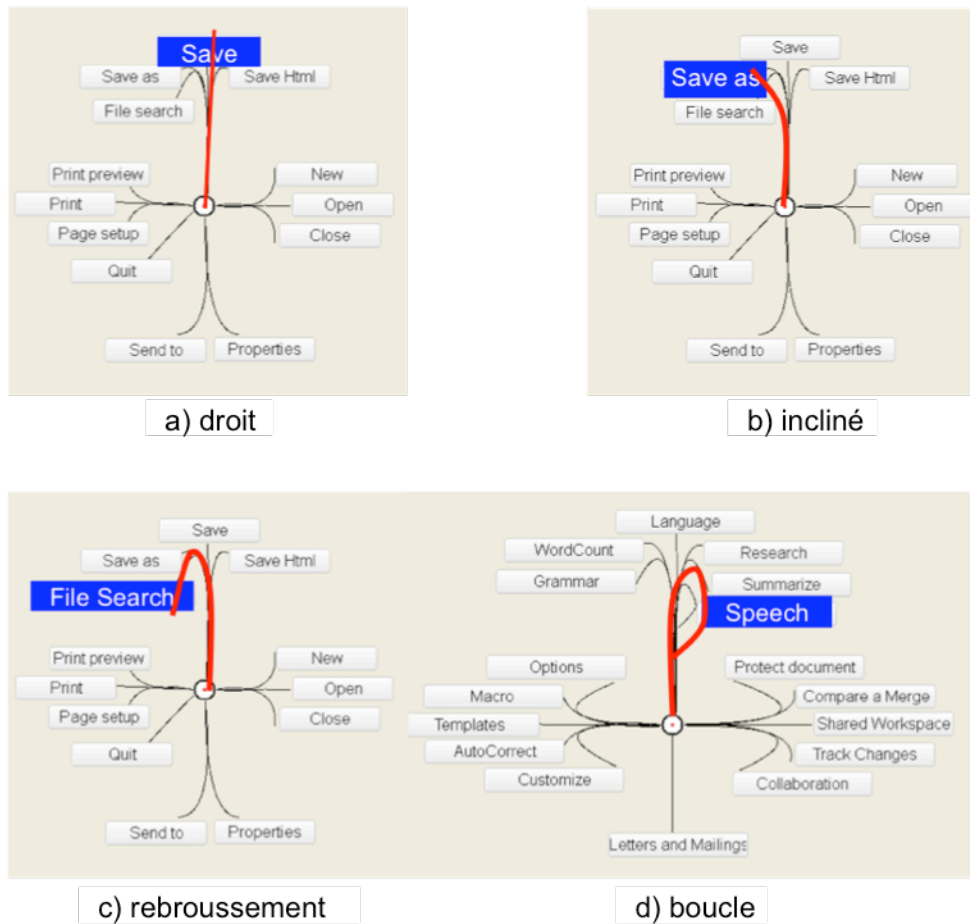


Figure 18 : 4 types de gestes utilisés dans les Flower menus. a) b) c) les menus contiennent les mêmes items que le menu « File » de Microsoft Word. D) Le menu contient les mêmes items que le menu « Outils » dans Microsoft Word.

Si l'utilisateur n'attend pas le délai, le menu fonctionne en mode expert et n'apparaît donc pas à l'écran. L'utilisateur active une commande en reproduisant la marque qu'il avait apprise en mode novice (Figure 19-b).

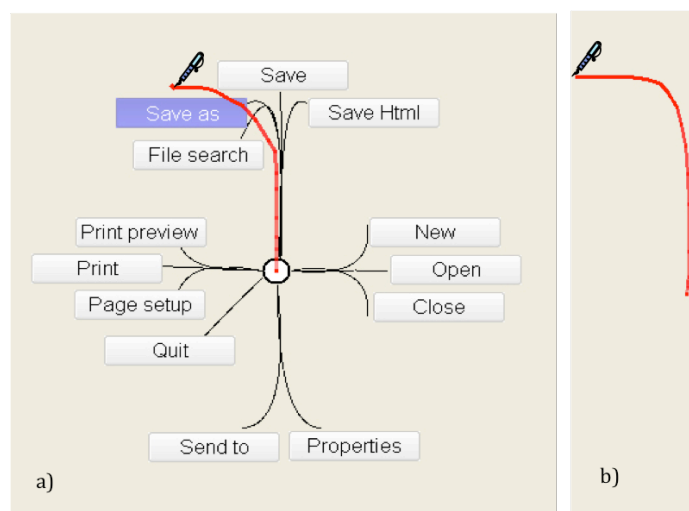


Figure 19 : a) Mode novice et b) Mode expert des Flower menus.

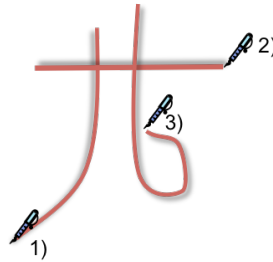


Figure 20 : Action en mode expert d'une commande de profondeur 3. L'utilisateur dessine 3 marques superposées (un geste courbe, un geste droit et une boucle).

Les Flower menus peuvent être hiérarchiques et fonctionnent alors comme les Multi-Stroke menus [Zhao 04]. En mode novice, les menus se superposent et en mode expert, l'utilisateur dessine une série de marques simples superposées comme celles illustrées à la Figure 20.

4.3 Propriétés des Flower menus

Nous décrivons maintenant les propriétés des Flower menus relativement à notre espace MenUA. Comme dans le chapitre précédent, nous abordons d'abord les propriétés associées aux critères d'utilisabilité, puis celles associées aux critères d'applicabilité.

4.3.1 Rapidité et précision

Activation du menu

Le Flower menu s'active, comme les Marking menus, lorsque l'utilisateur appuie sur le bouton droit de la souris.

Recherche visuelle

Nous avons rappelé au début de ce chapitre l'importance des groupements internes pour la recherche visuelle. Ils favorisent le balayage des commandes en particulier dans les longues listes. Dans les Flower menus, ils sont représentés par une branche (ou un pétale). Toutes les commandes d'un même groupement interne sont affichées selon la même orientation. Ceci a l'avantage qu'elles sont disposées dans la même région et activées par le même début de geste. C'est la fin du geste qui permet de les différencier grâce à la courbure et au sens.

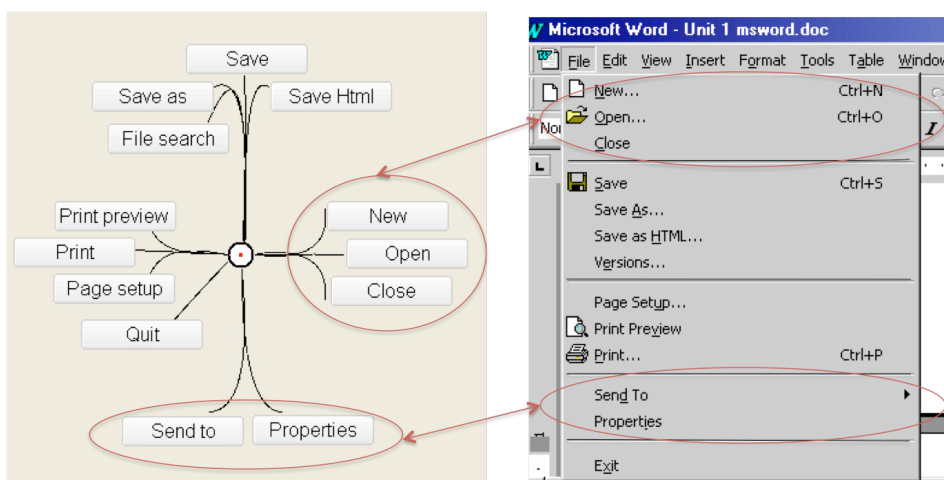


Figure 21 : Comparaison des groupements internes dans les Flower menus et dans les menus linéaires. Les commandes « New », « Open » et « Close » qui appartiennent au même groupement interne sont sur la même branche et partagent le même début de geste.

Les Flower menus permettent d'avoir jusqu'à 7 commandes par groupement interne : chaque orientation dispose de 7 gestes différents : S, B-, B+, C-, C+, P-, P+. Notre étude sur l'organisation des menus (section 1.3) montre que 7 gestes suffisent dans plus de 99% des cas. La seule exception identifiée est Photoshop qui dispose d'un menu avec une très longue liste de filtres. Les possibilités offertes par les groupements internes des Flower menus sont donc compatibles avec des configurations réalistes d'applications existantes.

Lorsque le nombre de commandes est trop important, les groupements internes ne sont plus suffisants et il devient nécessaire d'utiliser une structure hiérarchique. Les Flower menus autorisent à la fois les groupements internes et les groupements hiérarchiques. Dans leur version initiale, ils ne permettent pas la prévisualisation. Cependant, il est possible d'utiliser le mécanisme des Wave menus présenté au chapitre précédent (Figure 22). Un risque est d'avoir une représentation graphique surchargée (les anneaux + les marques + le libellé de tous les items).

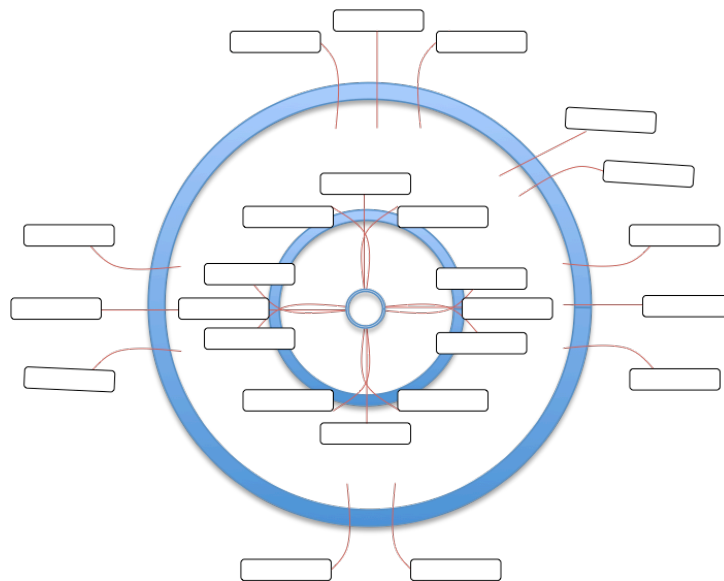


Figure 22 : Un Flower menu hiérarchique utilisant le mécanisme du Wave menu.

Sélection d'un item

Dans la plupart des applications, la distribution des commandes n'est pas uniforme [Witten 04] et peut être modélisée par une distribution Zipfian³ [Zipf 49]. D'après les résultats de la section précédente, les commandes fréquentes sont placées préférentiellement sur les gestes droits ou peu courbés.

Cette propriété diffère des Polygon et Zone menus pour lesquels tous les gestes sont de même nature ce qui empêche d'avoir des commandes privilégiées. De plus, Zhao [Zhao 06] a montré que les gestes du Polygon menu sont plus longs à exécuter que les gestes droits car ils nécessitent un tap supplémentaire. Ainsi, les commandes associées aux gestes droits des Flower menus sont plus rapides d'accès que toutes celles des Polygon menus.

Comme pour les Marking menus classiques, les gestes des Flower menus sont indépendants de l'échelle car la taille n'influe pas sur le système de reconnaissance.

³ Dans une distribution Zipfian, la probabilité d'occurrence d'un élément est très forte au départ et s'amenuise très vite. En d'autres termes, seulement quelques éléments apparaissent très souvent. Formellement, $P_n \sim 1/n^a$, où P_n est la fréquence d'occurrence d'un item au $n^{\text{ième}}$ rang. a est proche de 1. Par exemple, la distribution de 5 commandes pour $a=1$ est : 44%, 21.9%, 14.6%, 10.9% et 8.6%

Mode expert

La précision des Flower menus en mode expert ne devrait pas dépendre de la profondeur du menu car les marques sont composées temporellement (plutôt que spatialement) [Zhao 04].

4.3.2 Apprentissage et mémorisation

Apprentissage de la technique

L'augmentation du nombre de commandes nécessitent de « complexifier » les gestes (en considérant la position relative pour les Polygon menus ou la courbure et le sens pour les Flower menus) pour élargir le vocabulaire d'entrée. Cette hausse de la « complexité » gestuelle a inévitablement un impact sur la prise en main de la technique. Par exemple, les Polygon menus sont difficiles à prendre en main (section 5.5). D'une part, ils nécessitent deux actions (un tap + une marque) et les marques ne sont pas radiales (qui partent du centre du menu) à la différence d'autres types de Marking menus [Kurtenbach 91, Bailly 07, Zhao 04]. Nous détaillons ce point dans la section 5.5.

En considérant la courbure et le sens, les gestes des Flower menus restent radiaux et ne nécessitent qu'une seule action qui combine direction, courbure et sens en parallèle ; l'utilisateur presse le bouton de la souris, dessine la marque, puis relâche le bouton de la souris. Les utilisateurs ont dit préférer « les gestes partant du centre » avec les Flower menus plutôt que « de réaliser deux opérations avec les Polygon menus » durant notre évaluation présentée section 5.

Finalement, les Flower menus sont compatibles avec la forme la plus classique d'interaction : celle basée sur le clic sur les objets. Lorsque le curseur est au dessus d'un item, celui-ci est sélectionné (quel que soit le geste qui a été effectué préalablement). En mode novice, au relâchement du bouton de la souris, le système de reconnaissance considère la position du curseur si celui-ci se trouve au dessus d'un item. Si le curseur est en dehors d'un item, il interprète alors la marque dessinée. Ainsi, un utilisateur inexpérimenté peut se laisser diriger par l'affordance d'un item en cliquant dessus. Il n'aura alors aucune difficulté à prendre en main le Flower menu. Un retour-visuel (feedback) de la marque adéquate est affiché (quelque soit le type d'interaction) après l'activation d'un item afin d'inviter l'utilisateur à utiliser les gestes.

Apprentissage et mémorisation de l'organisation

La section 1.1 rappelait que les organisations en largeur aident l'utilisateur à construire rapidement une représentation mentale de la hiérarchie. Les Flower menus, en augmentant le nombre de commandes à chaque niveau, diminuent la profondeur du système de menus et favorisent ainsi la compréhension de la structure interne.

Une propriété importante des Flower menus est qu'ils offrent une grande souplesse aux concepteurs d'application pour l'organisation des commandes (en largeur ou en profondeur selon les besoins). N'étant pas limité par des contraintes techniques, ceux-ci peuvent adopter la structure qui leur semble optimale.

Finalement, les Flower menus mettent en valeur les groupements internes qui sont codés visuellement par une branche. L'utilisateur peut prédire le geste associé à une nouvelle commande à partir des autres commandes du même groupement interne. En particulier, le début du geste est le même pour toutes les commandes du groupe. Un autre point important est que la courbure peut jouer un rôle de « modifieur ». Par exemple, la commande « Save as » est une légère « modification sémantique » de la commande « Save ». Cette légère modification est codée au niveau du geste et de sa représentation. La commande « Save as » a le même geste que la commande « Save » mais avec une courbure légèrement plus accentuée. Le degré de courbure

traduit alors la distance sémantique entre les commandes. Ce codage permet à l'utilisateur d'apprendre plus facilement l'organisation des commandes.

Apprentissage et mémorisation du mode expert

Les Flower menus permettent une transition fluide du mode novice au mode expert, car l'utilisateur exécute les mêmes gestes dans ces deux modes de fonctionnement. L'utilisateur apprend implicitement le mode expert par la répétition du mode novice.

La section 5 montre que pour deux techniques (Flower et Polygon) qui ont les mêmes gestes dans les deux modes de fonctionnement, les utilisateurs n'acquièrent pas le mode expert à la même vitesse : après 5 minutes d'apprentissage, les utilisateurs ont appris deux fois plus de gestes avec le Flower qu'avec le Polygon. Ceci suggère que la répétition n'est pas le seul facteur pour l'apprentissage du mode expert. La nature des gestes et leur représentation dans le menu sont des facteurs sous-estimés pour l'apprentissage et la mémorisation qui n'ont peut être pas retenu une attention suffisante dans la littérature. Il est de fait vraisemblable que l'organisation radiale des Marking menus classiques et des Flower menus, leur « bonne forme » au sens de la gestalt [Helson 33] et la simplicité des gestes comptent pour beaucoup dans leur efficacité en ce qui concerne l'apprentissage. Nous développerons ce point dans la section 5.5.

4.3.3 Satisfaction

La prise en main de la technique a un impact important sur le plaisir que ressent l'utilisateur car c'est le premier contact avec la technique (section 3 du chapitre 4). Aussi, une attention particulière a été portée aux Flower menus pour qu'ils soient simples et qu'ils invitent les utilisateurs à les utiliser. Ceci contraste avec d'autres techniques, comme les Polygon menus, dont la configuration est efficace en mode expert, mais difficile à comprendre par des utilisateurs novices comme l'ont montré nos expérimentations.



Nous retenons pour les critères d'utilisabilité principalement le fait que les Flower menus améliorent la *recherche visuelle* et l'*apprentissage de l'organisation des commandes* grâce à leur flexibilité d'organisation et aux groupements internes. Ces derniers sont représentés par une branche (pétale) dont tous les gestes sont orientés dans la même direction.

Nous avons décrit les propriétés associées aux critères d'*utilisabilité* de notre espace MenUA, nous abordons celles associées aux critères d'*applicabilité*.

4.3.4 Adéquation à l'application

Les Flower menus ont été conçus pour contenir un nombre important de commandes (>1000). Ils sont hiérarchiques : ils peuvent avoir trois niveaux de profondeurs sans dégrader la précision grâce à la composition temporelle de marques simples [Zhao 04] (chapitre 5).

Un point important est le fait qu'ils augmentent significativement le nombre de commandes en largeur. Ils peuvent contenir en théorie $7 \times 8 = 56$ commandes à chaque niveau de profondeur. Dans la pratique, il n'est pas nécessaire d'avoir autant de commandes à un niveau donné mais ceci offre une grande souplesse pour organiser les commandes. Le Flower menu peut donc facilement contenir et organiser une vingtaine de commandes, un nombre suffisant pour la plupart des applications (la moyenne est 12.4).

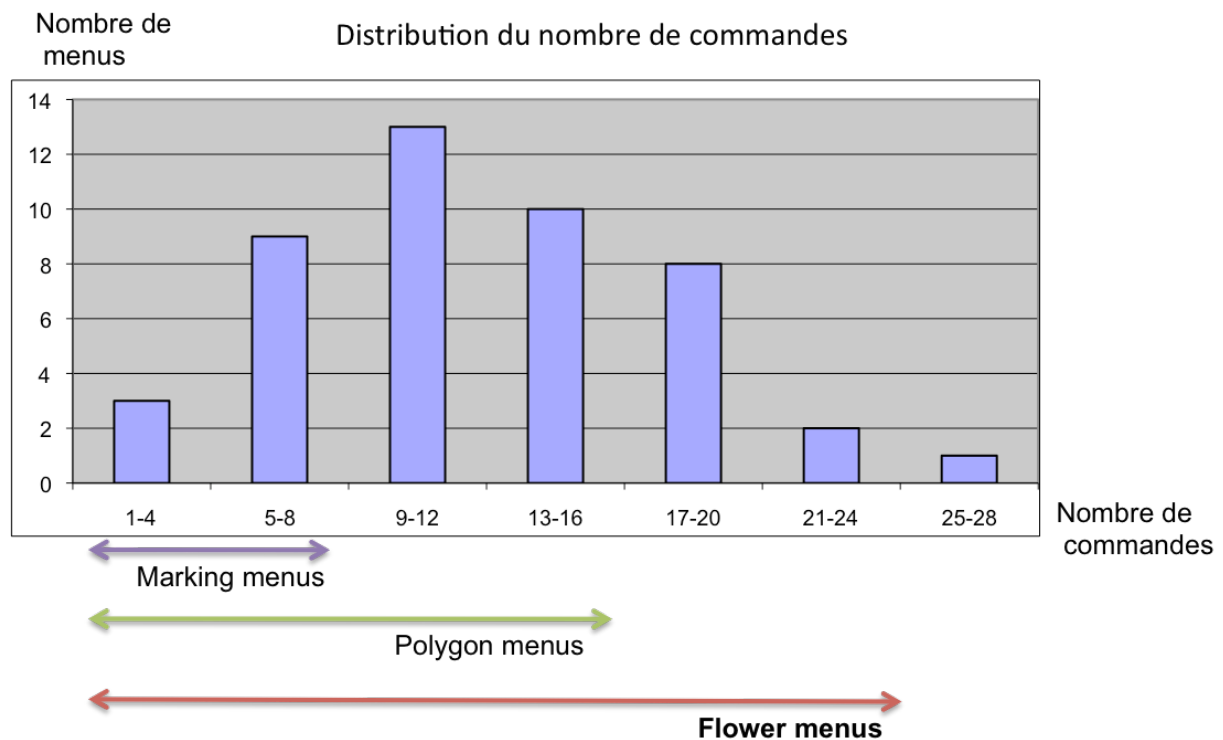


Figure 23 : Distribution du nombre de commandes dans les menus de 6 applications grand public.

4.3.5 Adéquation à la plateforme

En entrée, les Flower menus peuvent être utilisés avec tous les types de dispositifs de pointage 2D (directs ou indirects). Les gestes utilisés sont indépendants à l'échelle et peuvent être dessinés sur de petites surfaces comme des touchpads [Isokoski 02].

En sortie et en mode novice, comme tous les menus circulaires, les Flower menus nécessitent davantage de place (au moins deux fois plus) que les menus linéaires en largeur. Lorsqu'ils sont hiérarchiques, les Flower menus superposent les menus ou utilisent le mécanisme des Wave menus (chapitre 6) afin d'exploiter au mieux l'espace disponible.

4.3.6 Adéquation à la tâche

Transitions

L'encombrement des Flower menus restant raisonnable, il peuvent servir de menus contextuels et sont efficaces pour la *transition objet-commande* et la *transition objet-commande* (l'utilisateur ne s'éloignant jamais très loin du point d'activation).

Les Flower menus favorisent la *transition commande-valeur* en permettant de fusionner la sélection de commandes et le contrôle de valeurs via deux mécanismes détaillés ci-après.

Le mécanisme de « crossing » consiste à traverser un élément pour déclencher une action. Dans ce contexte, lorsque l'utilisateur dépasse une certaine distance seuil sur les gestes droits, la commande correspondante est sélectionnée et le contrôle de valeur commence. La Figure 24 illustre cette possibilité pour le contrôle du zoom. 4 commandes correspondent à des niveaux de zoom prédéfinies (0.25, 0.5, 2, 4) et une commande placée sur un geste droit permet d'ajuster le niveau continûment. Le seuil d'activation est figuré sur l'interface par une ligne arrondie. Une fois le seuil atteint (b), le menu disparaît (c) et la distance parcourue contrôle directement la valeur du zoom (d-e). Ce mécanisme est similaire à celui proposé par les Control menus [Pook 00].

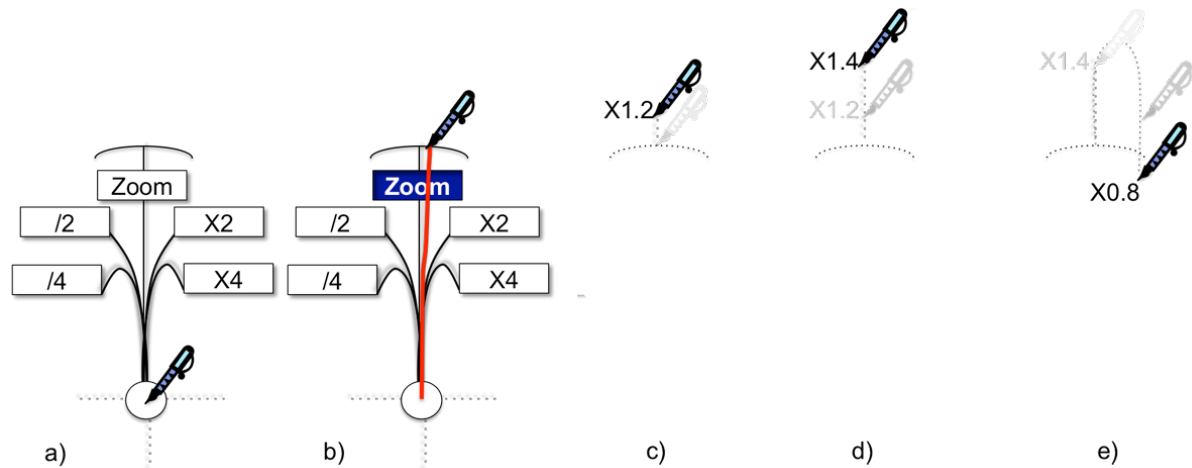


Figure 24 : Sélection d'une commande et contrôle d'une valeur par « crossing » sur les gestes droits. (a) l'utilisateur fait un geste en direction de la commande zoom. (b) Une fois, le seuil atteint, le menu disparaît (c). L'utilisateur contrôle alors la valeur du zoom (d-e).

Une technique alternative consiste à déclencher l'action non plus sur le franchissement d'un seuil mais lorsqu'une boucle (« pigtail ») a été refermée. Ce mécanisme est illustré à la Figure 25 où l'item "control", qui correspond à un geste boucle, déclenche le contrôle continu du zoom. Ce mécanisme s'apparente à celui des flowmenus [Guimbretiere 00] ou pigtail [Hinckley 05].

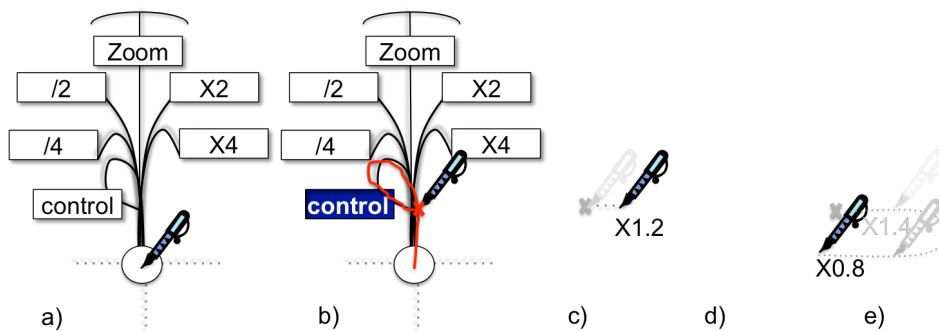


Figure 25 : Sélection d'une commande et contrôle de valeur par « fermeture » sur les gestes boucles. (a) l'utilisateur commence une boucle pour sélectionner le « contrôle de la précision du zoom ». (b) une fois, qu'il a fermé la boucle, le menu disparaît (c). L'utilisateur contrôle alors le niveau de précision de la commande zoom (d-e).

Le contrôle de valeurs semble plus difficile sur les gestes inclinés et de rebroussements. L'utilisation d'un seuil risque en effet de déclencher une action alors que l'utilisateur souhaitait continuer sa courbure.

Contrairement au « merging » qui consiste à sélectionner une commande et contrôler une valeur dans le même geste, la combinaison de commandes consiste à activer plusieurs commandes en un seul geste [Apitz 04]. Ce mécanisme favorise la *transition commande-commande*. La Figure 26 illustre cette possibilité. Au lieu de faire deux gestes : le premier pour créer un "Ovale" et le second pour lui appliquer une couleur, l'utilisateur fait un seul geste, correspondant à la concaténation des deux marques qui permet d'exécuter simultanément les deux commandes.

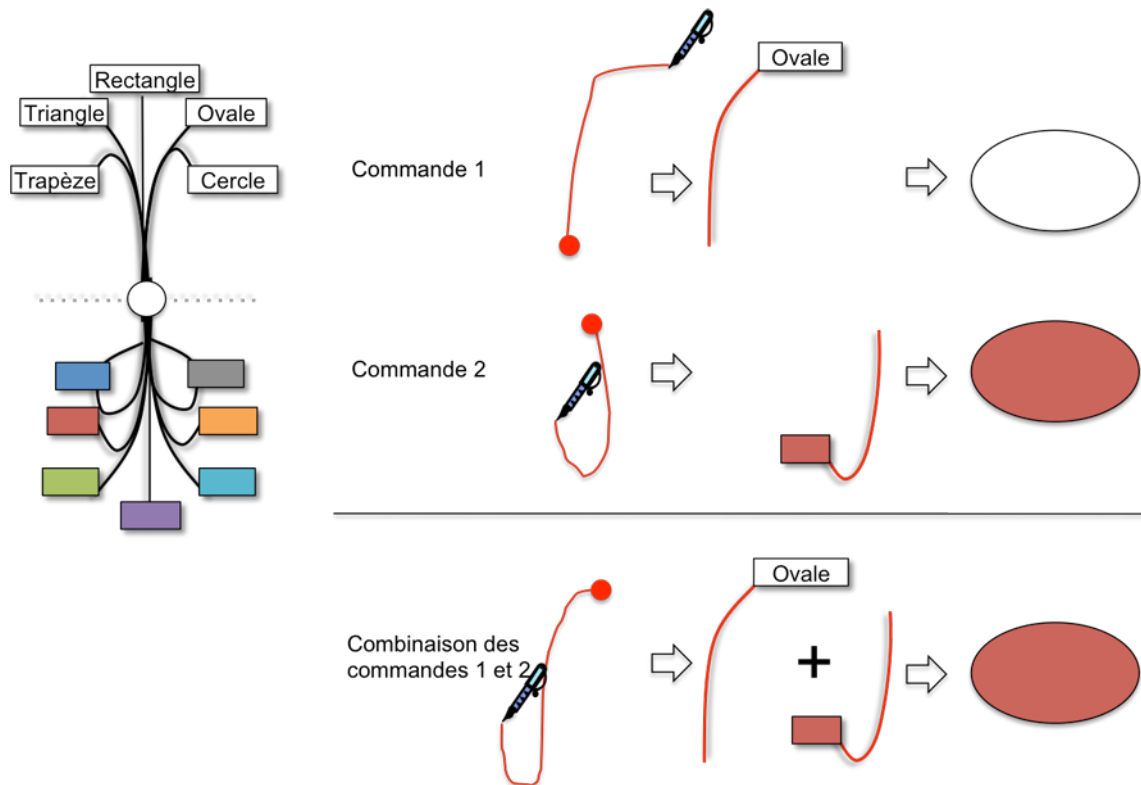


Figure 26 : Combinaison de commandes avec les Flower menus. En haut, l'utilisateur dessine deux marques séquentiellement pour créer un rectangle rouge (commande « Ovale » puis commande « rouge »). En bas, l'utilisateur combine ces deux commandes en un seul geste (1 marque au lieu de 2). Pour cela, il dessine la marque pour la forme ovale à l'envers qu'il concatène avec la marque pour la couleur rouge.

Contexte visuel

Afin de maximiser le contexte visuel de l'utilisateur, les Flower menus affichent uniquement les libellés des commandes (« labels only »). Comme les autres Marking menus, ils permettent la sélection sans regarder (« eye free sélection ») en mode expert. En évitant l'affichage des items, le mode expert permet aussi de ne pas masquer la zone de travail de telle sorte que l'utilisateur puisse rester concentrer sur sa tâche.



Nous retenons pour les critères d'utilité le fait que les Flower menus permettent d'avoir suffisamment de commandes en largeur pour répondre aux besoins de la plupart des applications.

Nous venons de présenter dans cette section une nouvelle technique de menus appelée Flower menus que nous avons décrite relativement à notre espace MenUA. L'originalité des Flower menus est d'avoir considéré la courbure et le sens en plus de l'orientation. Ces nouveaux gestes permettent d'avoir suffisamment de commandes en largeur pour répondre aux besoins de la plupart des applications. La représentation de ces gestes permet aussi de mettre en valeur les groupements de commandes en mode novice ce qui est utile pour la recherche visuelle et l'apprentissage.

5 Evaluation

La plupart des travaux sur les menus ayant mis l'accent sur la rapidité et précision, leurs évaluations portent également sur ce critère [Ahlström 05, Ahlström 06, Cockburn 06, Tsandilas 07, Tanvir 08] et [Kurtenbach 93, Balakrishnan 98, Zhao 04, Zhao 06]. Nous commençons par faire une revue des quelques travaux qui portent sur l'apprentissage du mode expert dans les menus et nous présentons ensuite notre expérience.

5.1 Expériences existantes

Les trois études expérimentales [Kurtenbach 94, Grossman 05, Cockburn 07] nous ont servi d'inspiration pour l'élaboration de notre expérience.

Une expérience longitudinale [Kurtenbach 94]

Kurtenbach [Kurtenbach 94] a étudié le comportement de deux participants qui devaient utiliser un Marking menu dans une tâche d'édition de grande quantité d'informations du logiciel ConEd. L'expérience s'est déroulée sur une dizaine d'heures réparties sur 30 jours. Les expérimentateurs enregistraient les sélections réalisées en mode novice et en mode expert. L'avantage de cette expérience est que les Marking menus sont placés dans un contexte d'application. Nous observons véritablement comment les participants utilisent les Marking menus. Les résultats ne sont pas biaisés par la tâche ou par les instructions. Malheureusement, ces expériences sont longues et difficiles à mettre en place en laboratoire ce qui explique le faible nombre de participants de cette étude.

Etude de l'apprentissage implicite [Grossman 07]

Grossman [Grossman 07] a réalisé une expérimentation contrôlée afin de comparer l'impact de 3 techniques d'interaction sur la rapidité d'apprentissage des raccourcis claviers des menus linéaires. Le véritable objectif de cette expérience n'était pas indiqué aux utilisateurs : la consigne était d'activer les commandes aussi vite et précisément que possible. Chaque participant testait une seule technique (plan inter-participants). Les expérimentateurs regardaient l'évolution du pourcentage de commandes sélectionnées correctement en mode expert à travers le temps. Nous notons deux inconvénients à cette expérience.

L'objectif de l'expérience est d'évaluer l'*apprentissage implicite* (apprentissage non dirigé), opposé à l'*apprentissage intentionnel* (forcé par l'expérience). Cependant, la notion d'apprentissage implicite n'est, par nature, pas clairement définie. Il est en effet difficile d'évaluer et de mesurer l'apprentissage implicite dans une expérience contrôlée car l'environnement ou les consignes de l'expérience peuvent influencer la manière dont les utilisateurs apprennent le mode expert. Par exemple, l'expérimentateur risque de susciter involontairement chez l'utilisateur l'envie d'utiliser le mode expert lorsqu'il lui explique son fonctionnement. Inversement, un manque d'explication peut inciter les participants à ne pas du tout utiliser le mode expert. Grossman [Grossman 07] a observé que les utilisateurs adoptent plusieurs stratégies pour l'apprentissage du mode expert. Certains (la moitié pour la technique de contrôle) n'ont pas du tout fait l'effort d'utiliser ce mode. Il devient alors difficile de distinguer l'apprentissage intentionnel de l'apprentissage implicite.

De plus, la durée de l'expérience est longue : une heure par technique. Les expérimentateurs ont donc du adopter un plan inter-participants qui a l'inconvénient d'augmenter la variabilité entre les participants et de nécessiter un plus grand nombre de participants (42 dans [Grossman 07]).

Etude de l'apprentissage intentionnel [Cockburn 07]

Dans [Cockburn 07], Cockburn s'intéresse à l'impact de deux techniques de visualisation pour l'apprentissage de la position des touches des claviers virtuels⁴. Comme pour les menus, nous pouvons considérer qu'il y a deux modes de fonctionnement : le mode novice où l'utilisateur dispose d'un retour visuel pour sélectionner une touche et le mode expert où l'utilisateur a appris la position de la touche et est capable de la sélectionner sans regarder. Les participants ont expérimenté les deux techniques (plan intra-participants) pendant environ 20 minutes. L'expérience est décomposée en trois phases : prise en main, entraînement et test.

La phase de prise en main (Familiarisation). Les différentes techniques sont expliquées aux participants et ils peuvent les tester pendant 50s.

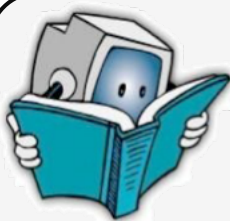
La phase d'entraînement. Les participants pouvaient s'entraîner pendant 5 minutes. Ils étaient informés que pour être le plus efficace pour la phase suivante, ils devaient mémoriser la position des touches. Il s'agit donc d'une tâche d'apprentissage intentionnel. Les symboles (stimulus) apparaissaient en haut de l'écran et les participants devaient retrouver la touche correspondante.

La phase de test. Les symboles de la phase d'entraînement sont réutilisés. Lorsqu'ils apparaissent en haut de l'écran, les participants doivent retrouver les touches correspondantes sans retour visuel (mode expert). Les participants n'étaient pas informés si la sélection était correcte ou incorrecte.

Les expérimentateurs mesurent ensuite le nombre de sélections correctes durant la phase de test. La technique qui a le nombre le plus élevé favorise l'apprentissage de la position des touches.

Cette évaluation comporte plusieurs avantages : elle se concentre sur l'apprentissage intentionnel lequel est plus facile à définir et à mesurer et elle peut facilement être mise en place en laboratoire. De plus, le plan est intra-participants ce qui permet de réduire la variabilité inter-participant. Nous avons adopté une approche similaire pour notre propre évaluation.

Récemment et après la publication des Flower menus, Bau [Bau 08] a proposé une expérience légèrement différente : son expérience comporte trois phases d'entraînement alternées avec trois phases de tests ce qui permet d'avoir une courbe d'apprentissage. De plus, il ne fixe pas le temps de la phase d'entraînement, mais le nombre de sélection effectué. Comme, l'apprentissage dépend à la fois du nombre de répétition et de la durée, il est difficile de conclure quelle approche est la meilleure.



Nous retenons que peu de travaux ont essayé d'évaluer l'apprentissage. Nous nous concentrons sur l'apprentissage intentionnel plutôt qu'implicite car ce dernier est difficile à capturer dans une expérience contrôlée.

⁴ La première technique de base consiste à afficher tous les symboles sur toutes les touches. La seconde technique consiste à afficher le symbole, seulement lorsque le curseur est au dessus de la touche. L'utilisateur est alors forcé de naviguer (« brush ») afin de trouver la touche désirée. Cette technique qui demande plus d'effort est supposée favoriser l'apprentissage de la localisation des touches.

5.2 Expérimentation

Notre expérimentation visait à comparer trois techniques de menus pour l'apprentissage intentionnel du mode expert : les Flower menus, les Polygon menus et les menus linéaires. Le menu linéaire sert de menu de référence. Les Polygon menus ont été choisis car ils peuvent contenir 16 commandes en largeur. Nous n'avons pas considéré d'autres techniques comme les Compound Marking menus [Kurtenbach 93] ou les Multi-Stroke menus [Zhao 04] car ces techniques ne permettent pas d'afficher suffisamment d'items à un seul niveau. Il aurait donc fallu introduire une nouvelle variable dans notre expérience : la profondeur. Au final, l'expérience aurait été trop longue.

Hypothèses

Nos hypothèses sont les suivantes :

H1 : *Les Marking menus (Flower menus et Polygon menus) favorisent la mémorisation du mode expert parce que les mêmes actions sont faites en mode novice et en mode expert.*

H2 : *L'apprentissage du mode expert est plus rapide avec les Flower menus qu'avec les Polygon menus. L'association geste – commande semble plus facile pour les Flower menus. Cette caractéristique peut favoriser l'apprentissage.*

H3 : *Les menus linéaires sont plus rapides que les Flower menus qui sont plus rapides que les Polygon menus en mode expert. Les raccourcis claviers nécessitent moins de temps que le dessin d'une marque [Odell 04]. La performance moyenne des gestes des Flower menus devrait être meilleure que celle des gestes des Polygon menus lorsque la configuration des Flower menus est bien-balancée, c'est à dire une configuration où les commandes sont placées préférentiellement sur les gestes droits et peu courbés.*

Nous décrivons maintenant les différentes configurations utilisées. Nous détaillons en particulier le nombre d'items utilisés, le nombre de groupements internes et leur taille. Enfin, nous justifions le choix des raccourcis claviers.

Nombre d'items

Il est courant de voir des menus avec au moins 16 items : 46% des menus ont plus de 14 items et 23% des menus ont plus de 16 items dans l'analyse présentée section 1.3. De plus 16 est le nombre maximal de commandes que peut avoir un Polygon menu. Nous avons donc choisi ce nombre pour l'expérience (Figure 28).

Groupements internes

Notre analyse montre que les groupements internes sont présents dans quasiment tous les menus linéaires et qu'ils contiennent généralement peu de commandes⁵. Nous avons donc adopté une configuration avec 7 groupements internes qui reflète ces statistiques :

- 2 groupes de 1 item ;
- 2 groupes de 2 items ;
- 2 groupes de 3 items ;
- 1 groupe de 4 items.

Cette configuration correspond à celle du menu « File » de Microsoft Word 2003 pour Windows (Figure 27).

⁵ 58% des groupements internes ont seulement 1 ou 2 items. 92% des groupements internes ont moins de 4 items.

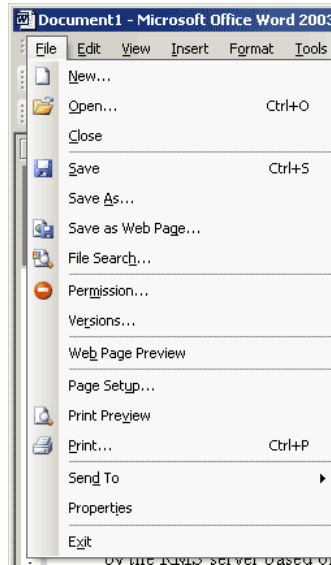


Figure 27 : le menu « file » de Microsoft Word 2003.

Les groupes les plus larges sont placés sur les axes principaux des Flower menus en accord avec les résultats de la première expérience section 3. Ils sont ordonnés de la même façon dans les trois menus, c'est à dire en commençant par le Nord-Ouest et en se dirigeant dans le sens anti-horaire pour les menus circulaires.

Les groupements sont implicitement visibles pour les Flower menus (branches) et les menus linéaires (séparateurs). Nous avons légèrement modifié la position des items du Polygon menu afin de révéler les groupements : les items appartenant à une même groupement sont plus proches. Au contraire, un espace plus important sépare deux groupements (Figure 28).

Les items d'un même groupement interne sont proches sémantiquement. Ils appartiennent à une même catégorie comme la couleur, animaux, musique, transports. Par exemple la catégorie sur l'enfance contient les termes Junior, Enfant, Bambin. La catégorie sur les matériaux contient Ivoire, Marbre et Argent. Les catégories et les noms des items ont été choisis de manière à ne pas introduire d'ambiguïtés. Un item ne pouvait pas appartenir à plusieurs catégories. Tous les items ont 6 lettres et ne contiennent pas de lettres rares comme Q, Z, Y, W, H.

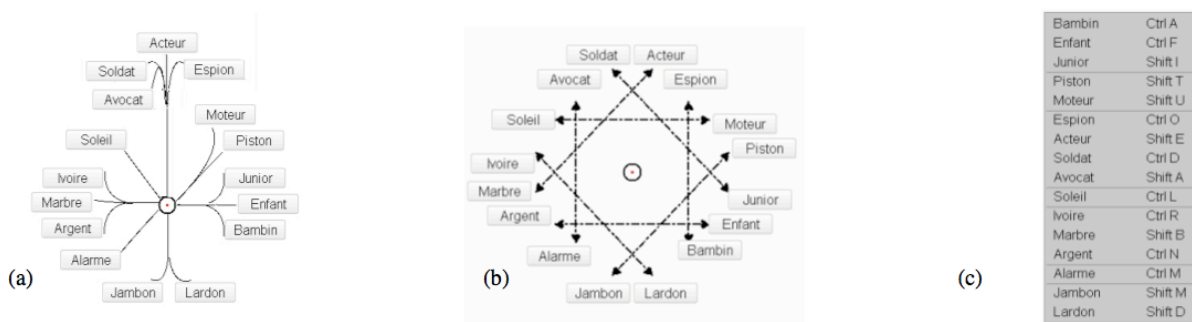


Figure 28 : Les trois menus utilisés dans notre expérience : (a) Flower menu, (b) Polygon menu, (c) menu linéaire.

Raccourcis claviers

Il existe une grande variété de raccourcis claviers dans les applications et ils ne sont pas toujours associés à une lettre du nom de la commande. On trouve par exemple, Ctrl+D pour « Police », Ctrl+F2 pour « Print preview ». Dans notre expérience, tous les raccourcis font partie du nom de la commande⁶ à l'exception de la première et de la dernière lettre afin d'éviter les singularités et de favoriser certains items. Nous avons exclu les raccourcis C, V, X et Z car ils risquaient d'introduire des variabilités indésirables. Nous avons en effet observé lors d'une expérience préliminaire que les participants développaient des stratégies spécifiques pour se souvenir de la relation entre l'item et son raccourci, probablement parce ces derniers sont très fréquents. Les modifieurs utilisés sont Ctrl et Shift. Nous n'avons pas considéré de combinaisons de modifieurs⁷. Une conséquence de cette configuration est que nous n'avons pas seulement utilisé les raccourcis situés sur la gauche du clavier comme c'était le cas dans les études précédentes [Grossman 07]. En effet, il n'était pas possible de trouver 16 raccourcis uniquement dans cette zone qui vérifiaient nos contraintes.

Stimulus

Le stimulus est le nom de l'item à sélectionner. Il est textuel plutôt qu'icnique afin d'éviter les confusions possibles entre les items proches sémantiquement. Il apparaît en haut de l'écran.

Nous n'avons pas utilisé une distribution Zipfian [Zipf 49], mais une distribution uniforme. Une distribution Zipfian ne pouvait pas être utilisée dans cette expérience car la mémorisation des items peut dépendre de l'ordre (pour les menus linéaires), de l'orientation (pour les menus circulaires) et du type de geste (pour les Flower menus). Il aurait alors fallu à la fois contrebalancer les fréquences et les types d'items pour pouvoir interpréter les résultats, ce qui aurait rendu l'expérience complexe.

Procédure

La procédure s'inspire de celle de [Cockburn 04] et comprend trois étapes : la prise en main, l'entraînement et le test.

Prise en main. Cette première étape consiste à expliquer le fonctionnement de la technique aux participants. Ces derniers testent le menu pour être sûrs qu'ils ont bien tout compris. Cette étape dure environ 2 minutes.

Entraînement. Nous indiquons aux participants le but de l'expérience (apprendre à sélectionner un maximum de commandes en mode expert) et leur expliquons qu'ils ne sont pas forcés de se presser car le temps n'est pas important dans cette phase. Un excès de rapidité dans la phase d'entraînement peut dégrader les performances dans la phase de test.

Les participants peuvent faire autant de sélections qu'ils le désirent pendant 5 minutes. Ils commencent en mode novice puis en mode expert lorsqu'ils sentent qu'ils sont capables de le faire. Le même item est présenté en cas d'erreur. Le stimulus était choisi aléatoirement avec la contrainte qu'il ne pouvait apparaître qu'une fois dans une séquence de 16 stimuli.

Test. Dans cette phase, les participants doivent sélectionner les items en mode expert correctement et aussi vite que possible. Le mode novice est désactivé. Les stimuli sont les mêmes que durant la phase d'entraînement et sont présentés aléatoirement. Cette phase est

⁶ Pendant la pré-évaluation, les raccourcis ne faisaient pas partie du nom de la commande comme dans l'étude précédente de Grossman [Grossman 07]. La plupart des participants se sont alors plaints de cette configuration, la trouvant trop difficile et trop éloignée des cas réels. Nous avons donc préféré utiliser des raccourcis appartenant aux noms des commandes afin de ne pas décourager les utilisateurs.

⁷ Les combinaisons de modifieurs sont souvent utilisés dans les applications, spécialement sur Macintosh ou l'on peut trouver : Shift + CMD + DEL, Alt + CMD + M, Alt + Shift + CMD + C.

répétée deux fois (2 blocs) afin de collecter plus de données expérimentales et pouvoir évaluer les performances en terme de rapidité. Dans cette phase, aucun retour visuel autre que la trace n'est donné aux participants. En particulier, ces derniers ne savent pas si la sélection est correcte ou pas. Cependant, une deuxième chance est donnée aux participants d'apprendre le menu entre les deux blocs : le menu est affiché pendant 15 secondes.

Design

L'ordre des trois techniques est contre-balancé entre les participants par un carré latin. Trois listes de noms (Figure 28) sont utilisées pour éviter les effets de transfert d'apprentissage entre les techniques. Comme ces trois listes ont été choisies pour être sémantiquement équivalentes, elles ne doivent pas avoir d'effet sur l'apprentissage. Cependant, nous avons également contre-balancé ces listes avec les techniques et l'ordre. Ainsi, toutes les techniques sont testées dans les mêmes conditions. Chaque participant réalise l'expérience en une session qui dure environ 40mn. En résumé, le plan expérimental est le suivant :

18 participants *
 3 techniques de menus *
 16 items *
 2 blocs
 = 1728 sélections

Participants et matériels

18 bénévoles (3 femmes, 15 hommes) entre 22 et 35 ans (moyenne 26 ans) ont été recrutés dans le personnel de l'Université (différents de ceux de l'expérience sur les gestes de la section 3) et ont reçu une poignée de bonbons pour leur participation. Ils sont tous droitiers et familiers avec les ordinateurs. L'expérience est réalisée dans les mêmes conditions que la première expérience (ordinateur Dell portable, souris, C++, Qt).

5.3 Résultats

Comme prévu, l'analyse de la variance ne rapporte aucun effet de *la liste d'items* sur la mémorisation ou la rapidité.

Apprentissage du mode expert

L'analyse de la variance montre que l'effet significatif principal est celui des *techniques* sur le nombre d'items rappelés en mode expert ($F_{2,34} = 70.34$, $p < 0.0001$). Un test de Tukey avec $\alpha = 5\%$ révèle (Figure 29) que les Flower menus, avec 81% d'items mémorisés (12.9/16) sont meilleurs que les Polygon menus (40%, 6.4/16) qui sont meilleurs que les menus linéaires (35%, 5.5/16). Les hypothèses H1 et H2 sont donc vérifiées, sauf que nous attendions une plus petite différence entre les Flower menus et les Polygon menus (car ce sont tous les deux des Marking menus) et une plus grande différence entre les Polygon menus et les menus linéaires.

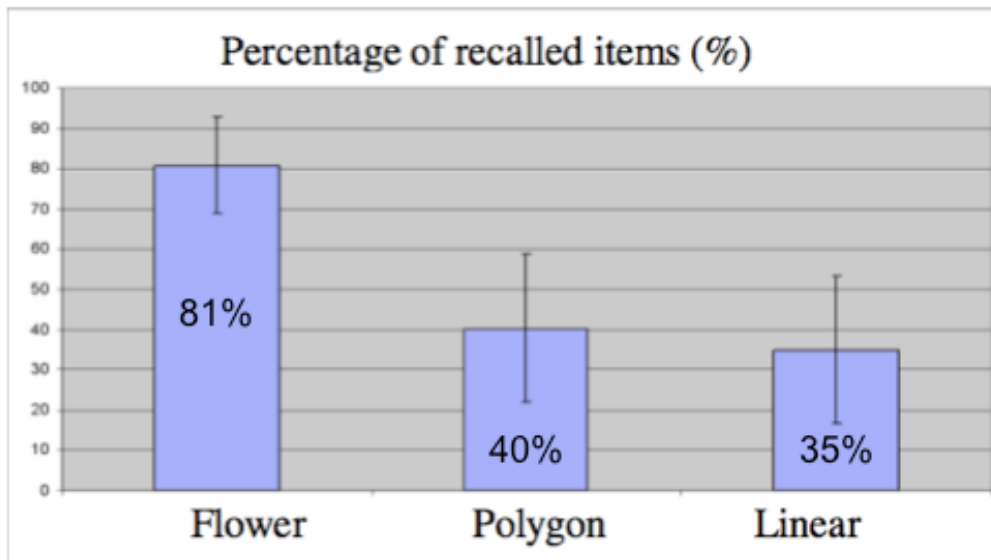


Figure 29 : Performance de la mémorisation du mode expert des commandes pour les trois techniques.

L'analyse de la variance rapporte également un effet de *l'ordre* sur le nombre d'items mémorisés en mode expert ($F_{2,34} = 5.69$, $p < 0.01$). Ce nombre est globalement plus important pour la technique testée en seconde position (9.5/16) que pour celle testée en première (7.3) et en troisième (8.0) position. Cependant, il n'y a pas d'effet pour l'interaction *technique*ordre*.

Rapidité

L'Anova indique un effet significatif pour les techniques sur le temps total ($F_{2,34} = 7.34$, $p < 0.01$). Un test de Tukey révèle que les Flower menus (2.4s) sont plus rapides que les menus linéaires (3.5s) et les Polygon menus (3.8s). L'hypothèse H3 n'est donc pas complètement vérifiée car les menus linéaires sont plus lents que les Flower menus.

Le temps total peut être décomposé en deux parties : le temps de réaction et le temps d'exécution. Le temps de réaction correspond à l'intervalle de temps entre l'apparition du stimulus et l'appui de l'utilisateur sur la souris ou le clavier. Le temps d'exécution correspond au temps pour dessiner la marque (Marking menus) ou d'appui sur les touches (menu linéaire).

L'Anova rapporte un effet des *techniques* sur le temps de réaction ($F_{2,34} = 9.0$, $p < 0.001$). Un test de Tukey indique que les menus linéaires (2.9s) sont plus lents que les Polygon menus (2.1 s) et les Flower menus (1.6 s). L' Anova indique également un effet significatif des *techniques* sur le temps d'exécution ($F_{2,34} = 21.58$, $p < 0.0001$). Un test de Tukey révèle que les menus linéaires (0.6 secondes) sont plus rapides que les Flower menus (0.8 s) qui sont plus rapides que les Polygon menus (1.7s). Les résultats des menus linéaires doivent être pris avec précaution car leur temps de réaction est surestimé et leur temps d'exécution sous-estimé. En effet, le temps nécessaire pour déplacer la main pour appuyer sur une touche devrait théoriquement être comptabilisé dans le temps d'exécution, mais ce n'était pas techniquement possible dans notre expérience.

Finalement, l'Anova indique aussi un effet significatif des blocs sur le temps total ($F_{1,17} = 14.4$, $p < 0.001$). Le bloc 2 (3.0s) est plus rapide que le bloc 1 (3.4s).

Préférence

Après l'expérience, les participants ont répondu à un questionnaire et classé les 3 techniques (Flower, Polygon et linéaire) par ordre décroissant. 17 participants sur 18 ont choisi le Flower menu comme leur technique préférée.

	Flower menus	Polygon menus	Linear menus
Prise en main	4.6	3.0	3.9
rapidité	4.3	3.1	3.9
Simplicité	4.5	3.1	3.4
Précision	3.7	3.9	4.8
Apprentissage	4.4	2.2	2.3
Fun	4.7	3.3	2.1

Tableau 3 : Synthèse des opinions des utilisateurs pour 6 critères avec une échelle de Likert à 5 points.

Nous leur avons aussi demandé leur opinion pour les critères suivants : prise en main, simplicité, mémorisation, rapidité, précision et "fun" avec une échelle de Likert à 5 points. Une analyse Anova suivie d'un test de Tukey montre que :

- Les Flower menus et les menus linéaires obtiennent de meilleurs résultats que les Polygon menus pour la prise en main (Flower : 4.6; Polygon : 3; Linéaire : 3.9), la rapidité (Flower : 4.3; Polygon : 3.1; Linéaire : 3.9) et la simplicité (Flower : 4.5; Polygon : 3.1; Linéaire : 3.4).
- Les menus linéaires (4.8) sont plus précis que les Polygon menus (3.9) et les Flower menus (3.7).
- Les Flower menus (4.4) semblent mieux favoriser l'apprentissage du mode expert que les Polygon menus (2.2) et les menus linéaires (2.3).
- Enfin, les Flower menus (4.7) apparaissent plus "fun" que les Polygon menus (3.3) eux mêmes plus "fun" que les menus linéaires (2.1).

La plupart des participants ont dit préférer les gestes « commençant au centre » avec les Flower menus plutôt que de « faire deux opérations » avec les Polygon menus (en faisant référence au tap + marque). La plupart des participants ont également souligné qu'il est plus facile d'apprendre et d'exécuter les gestes des Flower menus. Un participant résume l'avis général : « Je me dirige dans l'orientation générale de l'item, puis j'ajuste ».

Quelques participants ont trouvé « difficile d'apprendre 2 choses » avec les Polygon menus, « la position de l'item et le geste ». D'autres notèrent « qu'ils connaissaient la position de la commande, mais qu'ils étaient incapables de se rappeler le geste » avec le Polygon menu.

5.4 Discussions

Rapidité

Nos résultats sur le temps total indiquent que les Flower menus (2.4 s) sont plus rapides que les menus linéaires (3.5 s) et les Polygon menus (3.8 s). Une remarque importante est que notre expérience n'avait pas pour but d'évaluer les performances de rapidité/précision pour un usage expert, mais d'étudier la capacité des utilisateurs à apprendre le mode expert de ces menus. Les temps de réaction et d'exécution ne sont donnés qu'à titre indicatif mais ne doivent pas être interprétés comme des temps pour des utilisateurs experts du fait de la nature et de la courte durée de l'expérience. Les temps de réaction et d'exécution seraient alors plus courts. Par exemple, les temps d'exécution de l'expérience 1 sont 20% plus rapides et se rapprochent davantage d'un usage expert.

Mémorisation

Notre étude montre clairement que les Flower menus sont plus efficaces que les menus linéaires et les Polygon menus pour l'apprentissage des commandes en mode expert. Les Flower menus ont des résultats plus de deux fois supérieurs aux Polygon menus (12.9 vs. 6.4 items) qui sont eux-mêmes meilleurs que les menus linéaires (5.6 items). Comme, nous l'avons expliqué en début de section, il est important de rappeler que cette expérience s'intéressait à l'apprentissage explicite plutôt que l'apprentissage implicite. Il était demandé aux participants d'apprendre explicitement le mode expert. Il est cependant raisonnable de penser qu'une technique qui facilite l'apprentissage explicite favorisera également l'apprentissage implicite, surtout lorsque la différence est importante (plus de 2 fois si l'on compare les Flower menus aux menus linéaires). De plus, le fait d'avoir une interaction similaire en mode novice et expert ne peut que renforcer cet effet.

Il convient également de noter que les raccourcis claviers et les marques ne sont pas incompatibles. Bien que notre expérience compare les marques avec les raccourcis claviers, il est possible de combiner ces deux modes de fonctionnement pour le mode expert. Les raccourcis et les marques sont alors redondants et permettraient aux utilisateurs de choisir deux interactions différentes pour l'activation des commandes. Ainsi, l'introduction des Flower menus dans une application n'entrerait pas en conflit avec les habitudes usuelles. Les utilisateurs peuvent conserver les raccourcis claviers et apprendre progressivement les marques correspondantes. De plus, à l'image de certains raccourcis claviers qui sont partagés entre différentes applications (Ctrl + X, Ctrl + C, Ctrl + V), il serait intéressant de définir un vocabulaire de gestes normalisé. Dans cette optique, les gestes droits serviraient principalement pour les commandes les plus fréquentes. Par exemple, la commande « Save » pourrait être un geste vertical vers le bas, comme si on rangeait le document dans un coffre.

Bien que les résultats vérifient nos hypothèses (H1 et H2), ils ne correspondent pas exactement à ce que nous attendions au départ. En fait, comme les Flower et les Polygon menus sont tous les deux des Marking menus, nous pensions qu'il y aurait une plus petite différence entre ces deux techniques et une différence plus importante entre les Polygon menus et les menus linéaires. Une expérience plus longue aurait peut être abouti à des résultats différents mais rien ne permet de conclure en ce sens avec les résultats obtenus : une étude longitudinale serait nécessaire pour étudier plus précisément ces aspects.

La section suivante propose des explications sur les différences observées entre les Flower menus et le Polygon menus et qu'il serait utile d'étudier plus en avant. Aussi, ces explications sont présentées comme des pistes de réflexions.



Nous retenons que les Flower menus sont plus efficaces que les Polygon menus pour l'apprentissage du mode expert.

5.5 Apprentissage et mémorisation : pistes de réflexion

L'apprentissage et la mémorisation sont des notions complexes faisant intervenir plusieurs facteurs (répétition, motivation, etc.). Nous nous basons sur deux facteurs, l'indirection et le nombre d'actions à mémoriser, pour tenter d'expliquer les différences observées entre les Polygon et les Flower menus.

Les Polygon menus

Les Polygon menus introduisent une forte *indirection* entre la position d'une commande et le geste associé. Nous définissons comme indirection toute forme de détour forcé pour atteindre un élément. Les Marking menus classiques n'introduisent pas d'indirection pour accéder à une commande : le geste consiste à aller tout droit en direction de l'item. C'est un geste simple, naturel et élémentaire. Malheureusement, ces gestes (basés uniquement sur l'orientation) ne suffisent plus lorsque le nombre d'items est important (section 2.2), il semble donc nécessaire d'introduire une indirection. Dans le cas des Polygon menus, celle-ci est très importante pour trois raisons :

- L'orientation de la marque vs. l'orientation de l'item. La Figure 30 montre qu'il est nécessaire de faire une marque de l'est vers le nord pour accéder à un item situé au nord (item Nord-Nord-Ouest). L'orientation de la marque ne correspond donc pas à l'orientation de l'item.

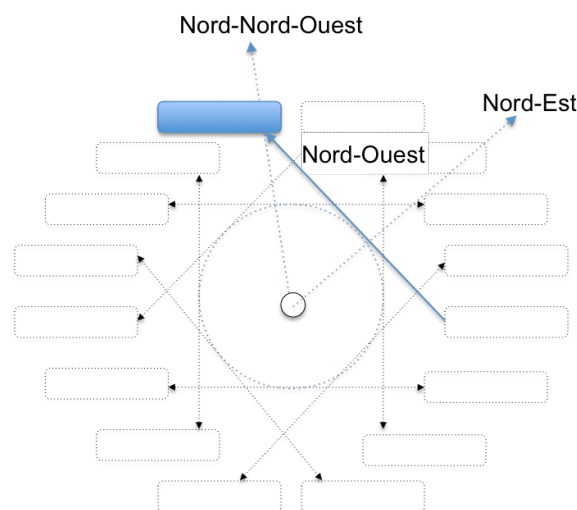


Figure 30 : Indirection entre l'orientation de l'item, de la marque et du décalage tangential dans les Polygon menus.

- La marque est tangentielle. Avant d'effectuer la marque, l'utilisateur doit se décaler. L'orientation de ce décalage ne correspond ni à celle de l'item, ni à celle de la marque. La Figure 30 montre que pour accéder à l'item (Nord-Nord-Ouest), l'utilisateur doit d'abord se décaler vers le Nord Est.
- Distance entre les items et les marque. La Figure 31-gauche montre que des marques proches (même orientation) sont associées à des items éloignés spatialement. Réciproquement, deux items proches spatialement ont des marques très différents (Figure 31-droite). Il semble difficile de prédire la marque d'un item à partir d'un item proche spatialement.

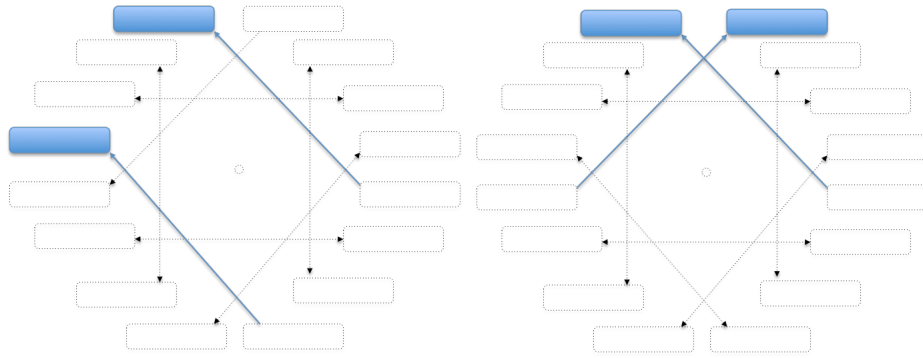


Figure 31 : A gauche, deux gestes proches (même orientation) donnent accès à deux items éloignés spatialement. A droite, deux items proches spatialement ont des gestes très différents.

Ces trois indirections permettent de mieux comprendre le ressenti des utilisateurs : « je connais la position de la commande, mais je suis incapable de me rappeler le geste ». L'association geste - item est difficile à assimiler car il y a trop d'indirections.

Il serait intéressant de vérifier ces hypothèses en changeant légèrement la représentation du Polygon menu afin de supprimer une partie de ces indirections sans modifier l'interaction. Par exemple, nouvelle représentation est illustrée Figure 32. Le mode expert serait exactement le même (un tap + une marque) et aurait donc les mêmes propriétés (indépendance à l'échelle, etc.). En mode novice, l'interaction serait également identique et la disposition globale des items reste inchangée. Les deux seules différences sont :

- la représentation des marques qui sont rapprochée du centre du menu ;
- l'affectation d'un item à une marque afin d'avoir un ensemble cohérent.

L'avantage de cette nouvelle représentation est que l'orientation d'un item correspond maintenant à l'orientation de la marque. De plus deux items proches spatialement sont associés à des marques proches (même orientation). Il serait intéressant de comparer ces deux représentations sur la même tâche d'apprentissage.

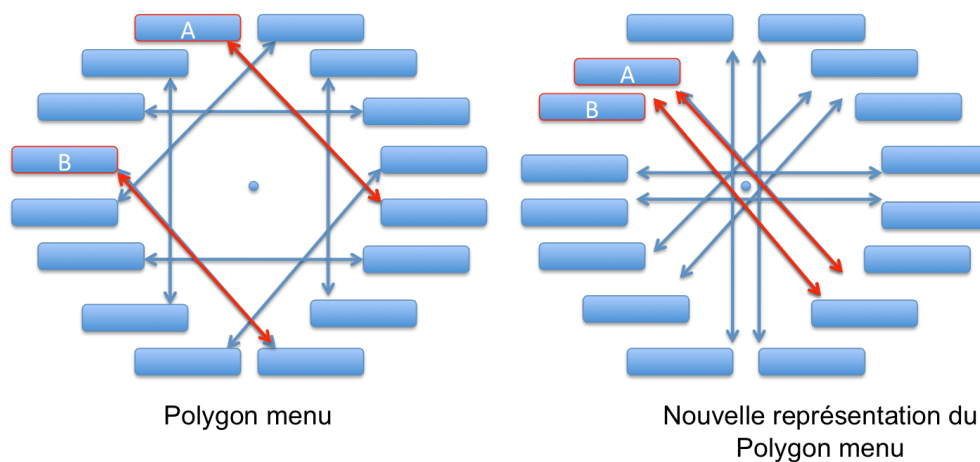


Figure 32 : A gauche, le Polygon menu. A droite, une nouvelle représentation du Polygon menu qui supprime une partie des indirections.

Différence avec les Flower menus

Cette nouvelle représentation du Polygon menu conserve cependant l'indirection associée au décalage tangentiel. Cette dernière est nécessaire pour pouvoir augmenter le nombre d'items en largeur. Nous constatons sur la Figure 33 que l'indirection est réalisée au début (décalage tangentiel) avec les Polygon menus et à la fin avec les Flower menus (courbure) :

- Dans les Polygon menus, l'indirection correspond au décalage tangentiel que l'utilisateur doit d'abord faire. La distinction entre deux items n'est pas faite à la fin du geste (même orientation), mais au début (décalage dans des sens opposés). La stratégie est donc d'aller du particulier au général
- Dans les Flower menus, l'indirection est à la fin. L'utilisateur part dans l'orientation de l'item, puis précise son geste en choisissant un sens lorsqu'il arrive près du but (à la fin de son geste). Ceci est en accord avec le ressenti des participants : « Je fais l'orientation générale, puis j'ajuste ». La stratégie est donc d'aller du général vers le particulier. Cette stratégie semble plus naturelle.

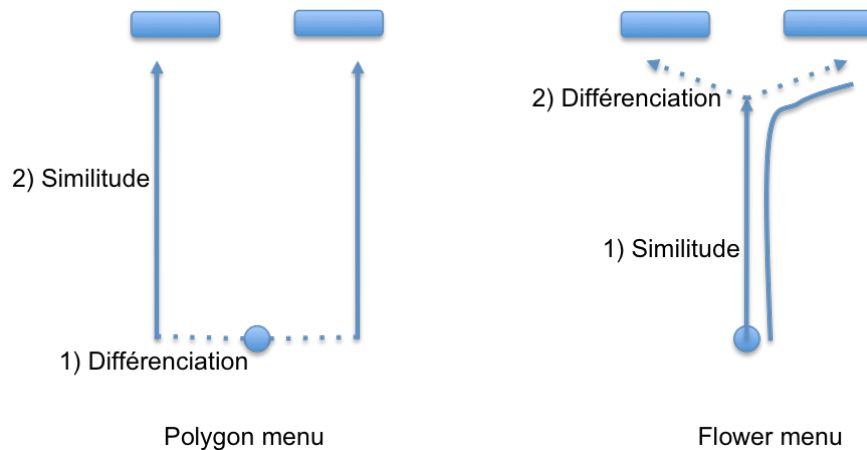


Figure 33 : Indirection au début avec les Polygon menus. Indirection à la fin avec les Flower menus.

Une autre différence entre ces deux techniques est que les Polygon menus impliquent deux actions consécutives (tap puis geste), tandis que les Flower menus ne nécessitent qu'une seule action (geste). Dans le cas des Polygon menus, ces deux actions correspondent à deux mnèmes (« chunk ») tandis que le geste à plusieurs dimensions (orientation, courbure, sens) des Flower menus peut correspondre à un seul mnème : les trois dimensions sont traitées en parallèle. Ce point pourrait expliquer pourquoi le Flower menu facilite l'apprentissage.

Efficacité des Marking menus

Nos observations invitent à reconsidérer les raisons de l'efficacité des Marking menus. L'argument souvent utilisé pour expliquer pourquoi les Marking menus sont meilleurs que les menus linéaires est que les utilisateurs apprennent le mode expert implicitement en répétant le même geste en mode novice. Cet effet est peut être surestimé puisque les Polygon menus, bien que basés sur ce principe de répétition, ont des performances finalement très similaires à celles des menus linéaires avec des accélérateurs claviers. Notre expérience suggère que d'autres effets entrent en jeu, en particulier le fait que certains gestes sont intrinsèquement faciles à mémoriser (gestes radiaux) et les regroupements facilitent l'apprentissage en permettant d'aller du général vers le particulier.



Nous retenons que les Flower menus introduisent moins d'indirection que les Polygon menus. De plus, Les Flower menus requièrent un geste unique tandis que les Polgyon menus impliquent deux actions consécutives. Ces deux constats réduisent la charge cognitive et semblent faciliter l'apprentissage du mode expert.

L'efficacité des Marking menus ne repose pas sur la répétition (transition fluide), mais aussi de la simplicité de ses gestes.

6 Conclusion

Face aux fortes contraintes des gestes des Marking menus classiques en ce qui concerne les organisations des menus en largeur, nous avons proposé un nouveau vocabulaire gestuel. Celui-ci exploite l'orientation, la courbure et le sens. Nous avons ensuite évalué la capacité des utilisateurs à dessiner correctement ces gestes lors d'une expérience contrôlée.

Nous avons proposé une nouvelle technique de menus, les Flower menus qui exploitent ce nouveau vocabulaire gestuel. Les Flower menus permettent d'augmenter fortement le nombre de commandes en largeur tout en garantissant de bonnes performances d'apprentissage.

Nous avons mis en évidence l'importance des groupements internes pour la recherche visuelle et l'apprentissage de l'organisation des menus. Non seulement, les Flower menus contiennent des groupements internes, mais leur représentation est particulièrement pertinente en les représentant visuellement par une branche et en les associant à un ensemble de gestes qui partagent les mêmes caractéristiques.

Nous avons mis en place, une évaluation visant à comparer l'impact de techniques d'interaction sur la rapidité d'apprentissage du mode expert. Elle a permis de comparer les performances des Flower menus, Polygon menus et Menus linéaires. Cette expérience a montré l'efficacité des Flower menus pour l'apprentissage du mode expert, non seulement par rapport aux menus linéaires, mais aussi par rapport aux Polygon menus bien que ces deux techniques reposent sur une transition fluide du mode novice au mode expert.

Face aux résultats décevants des Polygon menus, nous nous sommes interrogés sur les causes de leur manque de performances. Nous nous sommes intéressés aux facteurs, autres que la répétition, qui sont susceptibles de favoriser l'apprentissage. Le vocabulaire de gestes, leur représentation à l'écran, le lien avec les commandes, les indirections sont certainement des éléments qui interviennent dans l'apprentissage et la mémorisation des techniques d'interaction. Ces facteurs sont encore peu identifiés et rarement explicités dans la littérature. Une perspective prometteuse de ce chapitre consiste de mesurer ces facteurs pour en connaître l'importance de l'impact sur le critère *apprentissage et mémorisation*. L'importance de ce critère a été sous-estimée par rapport au critère rapidité et précision pour l'acceptation d'une technique de menus.

Petites et grandes surfaces interactives : Les Leaf menus et les Multi-Touch Menus (MTM)

Chapitre 8



1	LEAF MENU : UN MENU POUR DISPOSITIFS MOBILES	192
1.1	INTERACTION SUR DISPOSITIFS MOBILES	192
1.2	MENUS POUR DISPOSITIFS MOBILES	193
1.3	PRINCIPE DU LEAF MENU	194
1.4	PROPRIETES INTERACTIONNELLES DU LEAF MENU	196
1.4.1	RAPIDITE ET PRECISION	196
1.4.2	APPRENTISSAGE ET MEMORISATION	197
1.4.3	SATISFACTION	198
1.4.4	ADEQUATION A L'APPLICATION	198
1.4.5	ADEQUATION A LA PLATEFORME	198
1.4.6	ADEQUATION A LA TACHE	199
1.5	EVALUATION PRELIMINAIRE	199
1.5.1	EXPERIMENTATION	200
1.5.2	RESULTATS ET DISCUSSIONS	201
1.6	SYNTHESE	203
2	MULTI-TOUCH MENU : UN MENU POUR SURFACES MULTI-POINTS	203
2.1	DEGRES DE LIBERTE DE LA MAIN	203
2.2	SURFACES INTERACTIVES MULTI-POINTS	206
2.2.1	DEUX TECHNOLOGIES MULTI-POINTS	206
2.2.2	EXEMPLES DE TECHNIQUES D'INTERACTION MULTI-POINTS	207
2.3	TECHNIQUES DE MENUS MULTI-POINTS	209
2.4	MULTI-TOUCH MENU (MTM)	212
2.4.1	ACTIVATION DU MULTI-TOUCH MENU	212
2.4.2	SELECTION D'UNE COMMANDES AVEC LE MULTITOUCH MENU	215
2.5	PROPRIETES INTERACTIONNELLES DU MULTI-TOUCH MENU	216
2.5.1	RAPIDITE ET PRECISION	217
2.5.2	APPRENTISSAGE ET MEMORISATION	218
2.5.3	SATISFACTION	219
2.5.4	ADEQUATION A L'APPLICATION	220
2.5.5	ADEQUATION A LA PLATEFORME	220
2.5.6	ADEQUATION A LA TACHE	221
2.6	SYNTHESE	222
3	CONCLUSION	223

Dans les deux chapitres précédents, les techniques de menus proposées (Wave et Flower menu) n'étaient pas destinées à une plateforme spécifique et les évaluations expérimentales de ces menus ont été conduites sur ordinateurs personnels avec des dispositifs standard comme la souris. Dans ce chapitre, nous nous intéressons aux techniques de menus sur surfaces interactives considérant d'abord le cas des petites surfaces sur dispositifs mobiles puis celui des grandes surfaces comme celles des tableaux et tables augmentées. Dans le cas des petites surfaces, nous étudions les contraintes des dispositifs mobiles et leur impact sur la conception de techniques de menus tandis que dans le cas des grandes surfaces, nous visons à tirer partie des spécificités des dispositifs afin de concevoir des menus efficaces. Ce chapitre s'appuie donc sur le critère *adéquation à la plateforme* déjà caractérisé par dans MenUA au chapitre 5, section 2.

La section 1 est consacrée aux techniques de menus sur dispositifs mobiles. Nous exposons d'abord les contraintes associées à ce type de surfaces interactives (taille de l'écran, mode d'entrée limité, etc.). Nous présentons ensuite deux techniques de menus existantes conçues spécifiquement pour les dispositifs mobiles. Nous proposons alors une nouvelle technique de menus, les Leaf menus, qui enrichissent les menus linéaires de raccourcis gestuels. Les Leaf menus sont particulièrement adaptés à la sélection près des bords de l'écran. Enfin, nous concluons cette section par une évaluation préliminaire des Leaf menus.

La section 2 est dédiée aux techniques de menus sur de larges surfaces multi-points. La technologie multi-points offre de nouvelles ressources interactives permettant d'exploiter davantage les possibilités de la main. Nous décrivons d'abord un modèle de la main comportant 21 degrés de liberté. Nous présentons ensuite les rares techniques de menus existants sur tables interactives. Nous proposons alors, le Multi-Touch Menu, une technique de menu qui exploite les cinq doigts de la main.

1 Leaf menu : un menu pour dispositifs mobiles

Nous présentons le Leaf menu [Bailly 08], une nouvelle technique qui enrichit les menus linéaires de raccourcis gestuels. Le Leaf menu répond à plusieurs exigences associées à la sélection de commandes sur de petites surfaces interactives comme celles des dispositifs mobiles. Nous rappelons d'abord les difficultés liées à l'interaction sur dispositifs mobiles comme des pré-requis à la conception du Leaf menu. Nous présentons ensuite les menus existants pour ce type de dispositifs. Enfin, nous décrivons le principe et les propriétés interactionnelles du Leaf Menu.

1.1 Interaction sur dispositifs mobiles

L'interaction sur dispositifs mobiles est soumise à des contraintes spécifiques qu'il est nécessaire de considérer lors de la conception d'une nouvelle technique de menus [Karlson 07, Roudaut 08] :

- la petite taille de l'écran tactile ;
- des modes d'entrée différents et limités (stylet, clavier réduit ou inexistant) ;
- le contexte d'utilisation en situation de mobilité.

Ainsi, Pascoe montre dans [Pascoe 06], que les utilisateurs tiennent majoritairement leur dispositif mobile à une main en interagissant avec le pouce. Ce type d'interaction introduit de nouvelles contraintes par rapport à l'interaction au stylet [Karlson 07, Roudaut 08] :

- *L'occultation* : le pouce cache une partie de l'interface et les petites cibles sur lesquelles il est posé.
- *L'accès aux bords* : les bords opposés aux pouces sont parfois difficilement accessibles à cause de la morphologie de la main. De plus, les bords de l'écran peuvent être épais et gêner le positionnement du pouce près des bords.
- *L'imprécision* : le pouce étant plus large et plus instable qu'un stylet, il est plus difficile de pointer précisément. De plus, il est difficile pour l'utilisateur de prédire la position du pointeur sous son doigt.

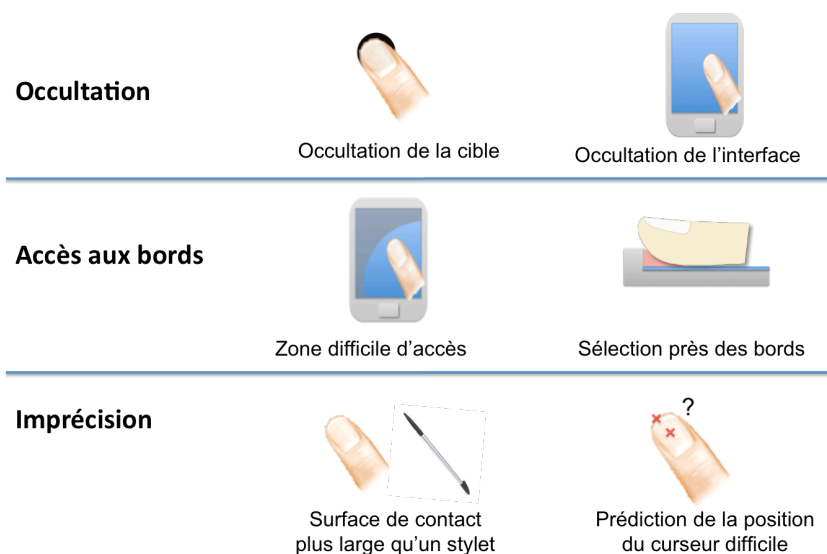


Figure 1 : Contraintes associées à l'interaction au pouce sur dispositif mobile. Illustration extraite de [Roudaut 08]

Alors qu'il existe de nombreux travaux pour améliorer le pointage d'une cible avec le pouce (« offset cursor » [Potter 88], Shift [Vogel 07], Thumbspace [Karlson 07], MagStick [Roudaut 08], TapTap [Roudaut 08] et Escape [Yatani 08],) il existe, à notre connaissance, uniquement deux menus conçus pour dispositifs mobiles : l'ArchMenu et le ThumbMenu [Huot 07] (section 2 du chapitre 5).

1.2 Menus pour dispositifs mobiles

ArchMenu [Huot 07]

ArchMenu [Huot 07], illustré Figure 2, est un menu semi-circulaire qui prend en compte la morphologie du pouce. Il s'active dans le coin inférieur droit. L'ArchMenu repose sur le mécanisme de la *projection de curseur* (« offset cursor »). Nous rappelons que ce mécanisme consiste à afficher le curseur à une certaine distance du pouce. Il évite que le curseur soit masqué par le pouce. De plus, il facilite les sélections sur le relâchement grâce à un filtre "moyenneur". La projection de curseur améliore donc la précision, mais s'avère être une technique relativement lente [Vogel 07, Roudaut 08].

ArchMenu est hiérarchique. Les sous-menus se déploient en éventail lorsque le curseur est au dessus d'un item hiérarchique (Figure 2). Cette organisation évite d'occuper trop d'espace écran en largeur contrairement aux Compound Marking menus [Kurtenbach 93a].

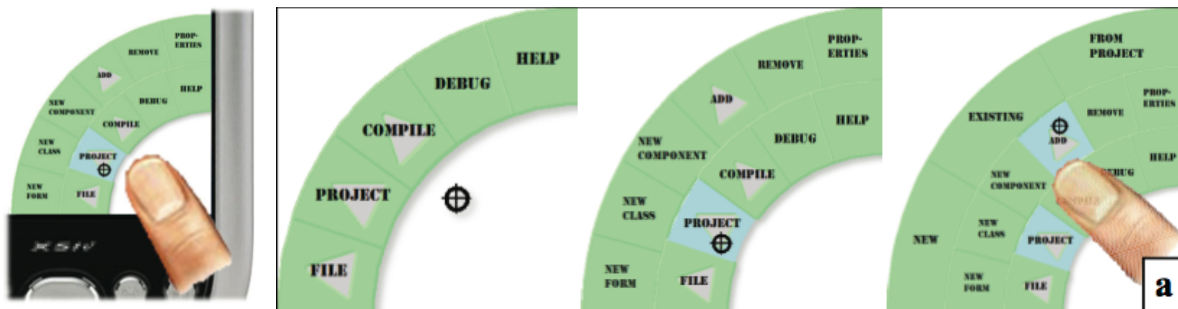


Figure 2 : ArchMenu [Huot 07]. Menu semi-circulaire qui s'active dans le coin en bas à droite de l'écran avec le pouce. Illustration extraite de [Huot 07].

ThumbMenu [Huot 07]

Comme l'ArchMenu, le ThumbMenu [Huot 07], illustré Figure 3, est un menu hiérarchique semi-circulaire reposant sur le mécanisme de l'« offset cursor » et se déployant en éventail. La différence est que le ThumbMenu est un menu contextuel : il peut être activé partout sur l'écran et il s'affiche de telle sorte qu'aucun item ne figure sous le pouce. Ainsi, comme le montre la Figure 3, un ThumbMenu est représenté par un anneau dont il manque $\frac{1}{4}$ de cercle.



Figure 3 : ThumbMenu [Huot 07]. Illustration extraite de [Huot 07].

De plus, lorsqu'un sous-menu est affiché sur le côté (Figure 4-a), l'utilisateur peut revenir au centre du menu afin de basculer l'ensemble du système pour obtenir que le sous-menu soit dans le prolongement du pouce (Figure 4-b). Ce mécanisme de rotation évite les problèmes d'occultation plus fréquemment associés aux items sur le côté [Huot 07].

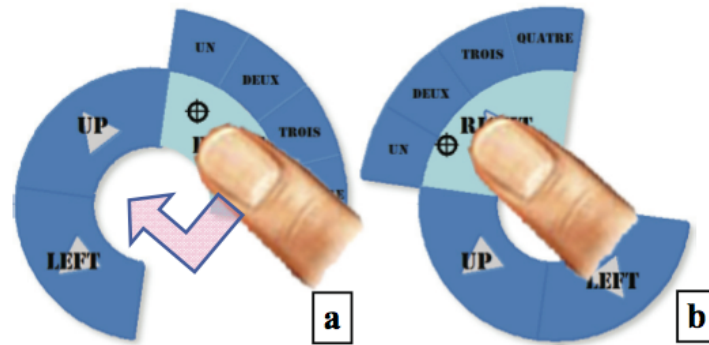


Figure 4 : Mécanisme de rotation du ThumbMenu. A) l'utilisateur revient au centre du menu afin que b) le système soit pivoté et le sous-menu affiché dans le prolongement du pouce. Illustration extraite de [Huot 07]

En résumé, ArchMenu et ThumbMenu [Huot 07] ont été conçus spécifiquement pour les dispositifs mobiles en prenant en compte la morphologie du pouce. Toutefois, ils ne proposent pas de mode expert. De plus, leur représentation semi-circulaire peut entraîner des problèmes de lisibilité. La Figure 2 montre par exemple qu'il est nécessaire de diminuer la taille de la police pour que le libellé puisse tenir dans l'item. Enfin, le ThumbMenu ne permet pas réellement les sélections près des bords de l'écran malgré le mécanisme de "basculement" des sous-menus. Le premier niveau de la hiérarchie peut en effet apparaître en dehors de l'écran ceci interdisant l'accès à certains items. Pour ces raisons, nous proposons dans la section suivante une nouvelle technique qui répond à ces limitations.

1.3 Principe du Leaf menu

Le Leaf menu, conçu en collaboration avec Anne Roudaut, est un menu linéaire enrichi de raccourcis gestuels [Kurtenbach 93, Appert 09]. Il est spécifiquement adapté aux petits écrans tactiles. Il dispose de deux modes de fonctionnement : le mode novice et le mode expert.

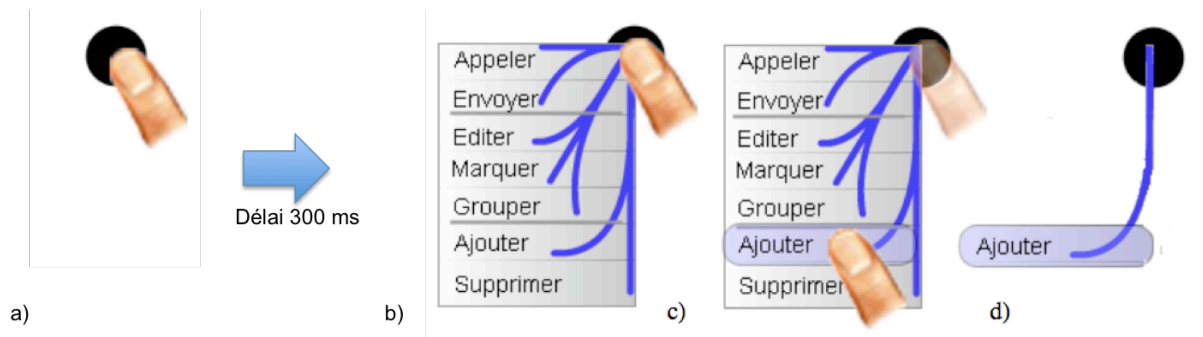


Figure 5 : Mode novice du Leaf menu : a) l'utilisateur presse une cible avec son pouce ; b) après un délai de 300ms, le menu apparaît. Une marque est associée à chaque item ; c) l'utilisateur relâche son doigt sur l'item désiré pour l'activer ; d) un retour visuel confirme l'item sélectionné et indique le geste à effectuer en mode expert.

Lorsque l'utilisateur patiente 300ms, le Leaf menu rentre en mode novice. Il fonctionne alors de manière similaire à un menu linéaire : les items sont affichés verticalement et sont activés sur le relâchement du doigt ou du bouton de la souris sur cet item (Figure 5). Cependant, il y a plusieurs différences majeures :

- Le menu s'affiche par défaut dans la zone sud-ouest (pour les droitiers) afin de minimiser l'occultation du doigt ou de la main sur le menu (Figure 5).
- Les items favoris peuvent être associés à une marque qui est affichée sur le menu. Cette marque servira de raccourci gestuel en mode expert. Dans le cas de la Figure 5, tous les items sont des favoris. Lorsqu'un item est activé en mode novice, un retour visuel confirme l'item sélectionné et montre la marque associée.

- Il utilise un effet *miroir* (Figure 6) lorsque l'espace sous le point d'activation n'est pas suffisant pour afficher le menu. Cet effet consiste à afficher le menu au dessus du point d'activation et à inverser l'ordre des items par symétrie verticale. Ainsi, l'item 1 de la Figure 6 est toujours l'item le plus près du point d'activation et l'item 7 le plus éloigné. Cette caractéristique a plusieurs avantages que nous expliquerons vis-à-vis des critères de l'espace MenUA dans les sections 1.4.1 et 1.4.5. Il convient de noter l'ordre des items n'est pas inversé lorsque le menu est affiché à droite ou à gauche du point d'activation.

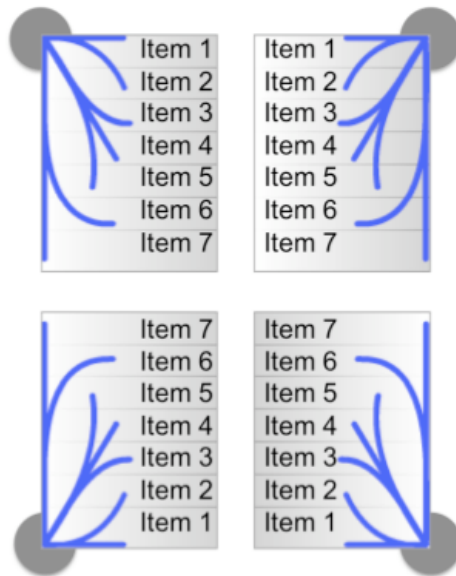


Figure 6 : Effet miroir du Leaf menu : la distance d'un item au point d'activation est invariante.

Lorsque l'utilisateur n'attend pas le délai, le menu est en mode expert : il n'est plus affiché et une commande peut être activée grâce à son raccourci gestuel (Figure 7). Une caractéristique intéressante du Leaf menu est que chaque item est associé à 4 raccourcis gestuels équivalents (Figure 6) du fait des symétries horizontale et verticale sauf l'item 1 et 7 où 2 des 4 gestes sont confondus. Par exemple, la Figure 8 montre les 4 gestes associés à l'item 6. Nous reviendrons sur les avantages de cette redondance gestuelle dans la section 1.4.5.

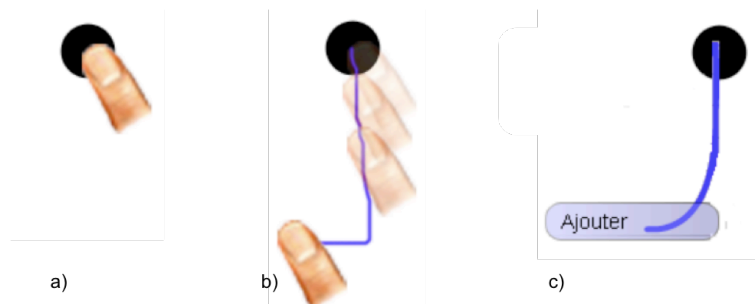


Figure 7 : Mode expert du Leaf menu : a) l'utilisateur presse une cible puis b) dessine directement la marque associée à l'item désirée. A la fin de son geste, un retour visuel apparaît pour confirmer l'item sélectionné.

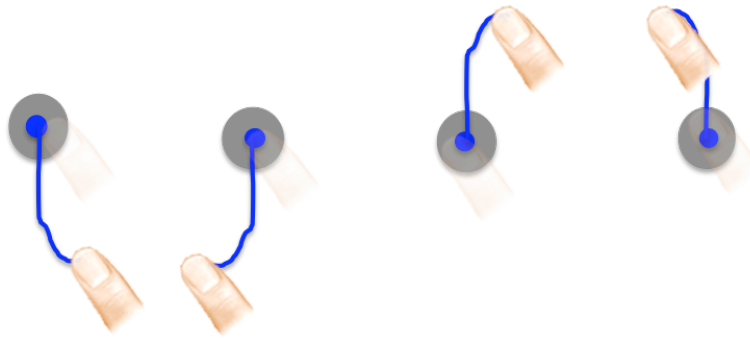


Figure 8 : Les différents raccourcis gestuels associés à l'item 6 ("Ajouter") de la Figure 6.

1.4 Propriétés interactionnelles du Leaf menu

Avant de présenter une évaluation expérimentale, nous décrivons plus en détail les propriétés du Leaf menu en nous basant sur notre espace MenUA. Nous présentons d'abord les propriétés associées au critère d'utilisabilité, puis celles associées au critère d'applicabilité.

1.4.1 Rapidité et précision

Activation du menu

Le Leaf menu est activé au contact du stylet ou du doigt sur la surface tactile.

Recherche visuelle

Un Leaf menu hiérarchique conserve la propriété de prévisualisation des menus linéaires. Lorsque le doigt de l'utilisateur est au-dessus d'un item hiérarchique, le sous-menu correspondant est automatiquement affiché, à gauche ou à droite de cet item en fonction de la place disponible. Ainsi, les utilisateurs peuvent facilement avoir un aperçu rapide des commandes des sous-menus.

Sélection d'un item

Les menus linéaires disposent généralement les items fréquents en haut du menu, près du point d'activation, afin que l'utilisateur puisse les sélectionner plus rapidement (Figure 9-a). Lorsqu'un menu linéaire est affiché trop près du bas de l'écran, il est translaté vers le haut afin que ses items restent visibles (Figure 9-b). Les items fréquents se retrouvent alors éloignés du point d'activation ceci forçant l'utilisateur à parcourir une distance importante pour les sélectionner. Au contraire, les Leaf menus conservent toujours les premiers items près du point d'activation grâce à l'effet miroir (Figure 6). Cette caractéristique a donc l'avantage de faciliter la sélection des items fréquents près des bords de l'écran, une situation courante sur les petits écrans.

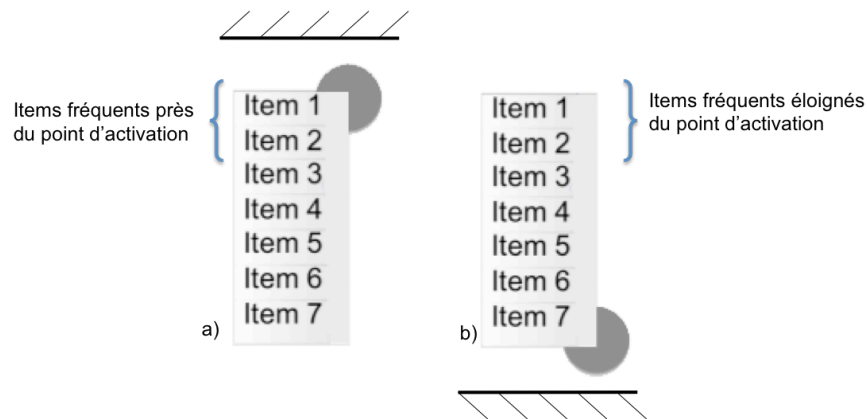


Figure 9 : Positionnement d'un menu linéaire et ordonnancement de ses items : a) le point d'activation est près du bord haut de l'écran ; b) le point d'activation est près du bord bas de l'écran.

Mode expert

Contrairement à l'ArchMenu, au ThumbMenu [Huot 07] et aux menus linéaires classiques, le Leaf menu dispose d'un mode expert reposant sur les raccourcis gestuels (Figure 7). Ces derniers permettent de palier l'absence de raccourcis claviers sur surface tactile. Sur les ordinateurs personnels, les utilisateurs peuvent choisir d'utiliser soit les raccourcis gestuels soit les raccourcis claviers : ce sont des modalités redondantes [Coutaz 94].

Utiliser uniquement des gestes droits ne permet pas de garantir un haut niveau de précision lorsque le menu contient 3-4 favoris. Pour cette raison, nous avons introduit certains gestes courbes du Flower menu (chapitre 7), plus précisément les gestes inclinés (« bent ») car ces sont les gestes les plus rapides et les plus précis après les gestes droits. Nous vérifions dans la section 1.5, que ces gestes sont suffisamment simples pour que l'utilisateur puisse les exécuter précisément avec le pouce.

Dans le cas des Leaf menus hiérarchiques, nous réutilisons le mécanisme des Multi-Stroke menus [Zhao 04] : l'utilisateur dessine plusieurs marques simples (une par niveau de la hiérarchie) qui peuvent être superposées (Figure 10).

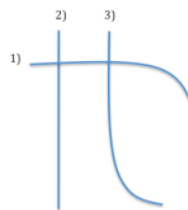


Figure 10 : Exemple de trace associée à un Leaf menu hiérarchique avec 3 niveau de profondeur.

1.4.2 Apprentissage et mémorisation

Technique

Le Leaf menu est une technique hybride qui fonctionne comme les menus linéaires en mode novice, mais qui permet comme les Marking menus, l'interaction gestuelle en mode expert. Ainsi, un utilisateur néophyte peut facilement prendre en main le Leaf menu car il ne sera pas surpris ni par sa représentation ni par son interaction en mode novice.

Organisation

Le Leaf menu permet d'avoir des groupements internes comme le montre la Figure 11. En plus des séparateurs, les marques renforcent l'appartenance des items à un groupe : les marques associées aux items d'un même groupe partagent la même orientation (une

caractéristique détaillée dans le chapitre 7 à propos des Flower menus). Lorsque le nombre d'items dans un groupement interne est trop important, seuls les items les plus fréquents sont associés à des raccourcis gestuels (Figure 11).



Figure 11 : Groupements internes dans les Leaf menus.

Mode expert

Le raccourci gestuel associé à un item ne correspond pas exactement au geste à faire en mode novice. De plus, à un item correspondent plusieurs raccourcis gestuels. Cependant, une évaluation, présentée dans la section 1.5, montre que la simplicité des gestes et leur corrélation avec la position d'un item dans le menu permettent d'apprendre rapidement le mode expert. Nous reviendrons sur ce point dans la section 1.5.2.

Une fois qu'une commande a été activée, le retour visuel en mode novice (Figure 5) et en mode expert (Figure 7) de la marque associée suggère à l'utilisateur de faire le geste correspondant [Tapia 95].

1.4.3 Satisfaction

Lors de notre évaluation (section 1.5), les utilisateurs ont apprécié le fait de pouvoir se passer du stylet pour interagir avec le menu. Ils ont également apprécié la sélection de commandes avec des gestes. Enfin, l'inversion du sens d'affichage ne semblait pas gêner les utilisateurs : au contraire, certains d'entre eux l'ont trouvé "plus logique".

1.4.4 Adéquation à l'application

Même si le Leaf menu est limité à 7 raccourcis gestuels, il peut contenir davantage de commandes en largeur : les raccourcis gestuels sont alors réservés aux commandes les plus fréquentes. Il permet également les organisations en profondeur car il est hiérarchique. De plus, le mode expert repose sur la composition séquentielle de marques garantissant un haut niveau de précision même avec 2 ou 3 niveaux de profondeurs [Zhao 04]. Le Leaf menu peut donc contenir un grand nombre de commandes quelle que soit la taille des libellés.

1.4.5 Adéquation à la plateforme

Même si le Leaf menu peut être appliqué aux ordinateurs personnels, il a été conçu à l'origine pour les dispositifs mobiles ayant des contraintes spécifiques (section 1.1) telles que la taille de l'écran ou l'imprécision de l'interaction au pouce.

Le Leaf menu est particulièrement adapté aux petits écrans. En mode novice, il peut s'afficher aussi bien à droite ou à gauche, en bas ou en haut du point d'activation (Figure 6). Ceci évite que le menu apparaisse en dehors de l'écran. En mode expert, chaque item dispose de plusieurs raccourcis gestuels (Figure 8). Cette redondance gestuelle permet de s'adapter aux contraintes spatiales. En effet, la Figure 12 montre qu'il est possible de sélectionner l'item n°6 dans les quatre coins de l'écran. C'est une différence essentielle avec les menus circulaires, dont une partie des items ne peuvent pas être sélectionnés en mode expert près des bords de l'écran : l'item positionné à l'ouest ne peut pas être sélectionné en mode expert près du bord gauche de l'écran).

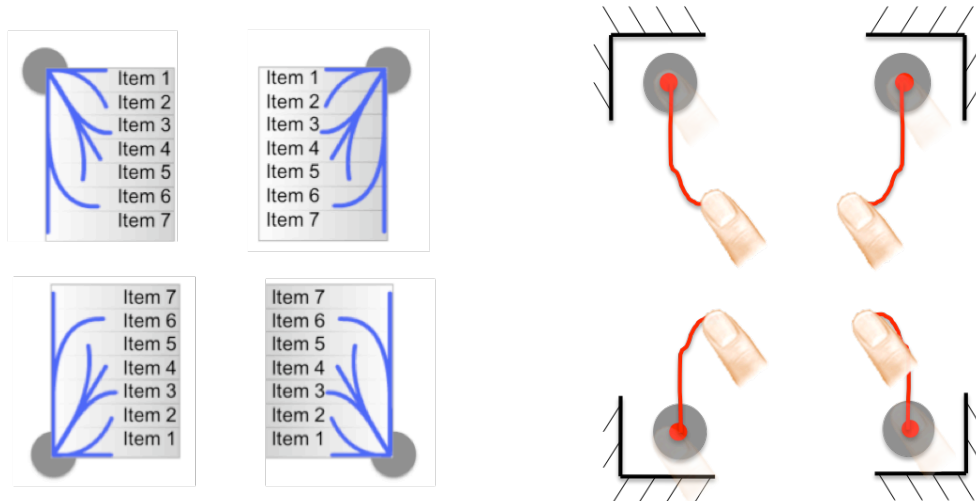


Figure 12 : Les 4 raccourcis gestuels associés à l'item 6 permettant la sélection de cet item en mode expert quelque soit l'endroit où se situe le point d'activation.

Sur écran tactile, la main de l'utilisateur peut masquer une partie des items des menus linéaires classiques car ils sont affichés dans la zone sud-est par défaut. Pour cette raison, le Leaf menu est affiché dans la zone sud-ouest [Hancock 04] lorsque l'espace disponible est suffisant.

Enfin, en augmentant la hauteur des items, le leaf menu évite les imprécisions généralement associées au relâchement d'un doigt sur écran tactile [Roudaut 08]. De plus, un filtre "moyenneur" a été utilisé afin de stabiliser le curseur et éviter les relâchements accidentels.

1.4.6 Adéquation à la tâche

Le Leaf menu, comme le ThumbMenu [Huot 07], est un menu contextuel qui peut être activé sur les objets d'intérêt favorisant la transition objet->commande. Cependant, lorsque les cibles sont trop petites, l'interaction au pouce peut s'avérer difficile. Il convient alors de combiner le Leaf menu avec des techniques pour améliorer la sélection de petites cibles comme TapTap [Roudaut 08]. De plus, contrairement aux menus circulaires, la représentation du Leaf menu (comme celle des menus linéaires), ne masque que partiellement les petites cibles. Ainsi, l'utilisateur peut vérifier qu'il a sélectionné la bonne cible.



Nous retenons que le Leaf menu est un menu linéaire enrichi de raccourcis gestuels particulièrement adapté aux écrans tactiles des dispositifs mobiles. Il permet la sélection d'items près des bords de l'écran en mode novice et expert. Son originalité réside dans l'introduction d'un effet miroir favorisant la sélection des items fréquents.

1.5 Evaluation préliminaire

Nous avons réalisé une évaluation informelle afin d'étudier :

- l'impact de l'effet miroir sur les utilisateurs ;
- leur capacité à dessiner correctement les marques du Leaf menu ;
- leur capacité à mémoriser les commandes en mode expert.

Nous présentons l'expérimentation puis nous analysons les résultats.

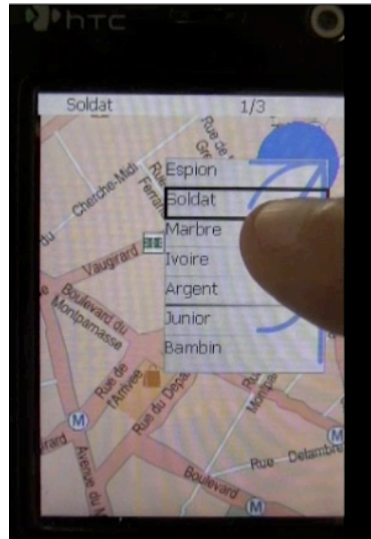


Figure 13 : Sélection d'un item au pouce avec le Leaf menu sur un PDA-Phone.

1.5.1 Expérimentation

Organisation du menu

Le menu Leaf contient 7 items (Figure 13) organisés en trois groupements internes de taille 2-3-2. Les noms des items dans un même groupe sont sémantiquement proches (professions, matériaux, enfance)

Stimulus

Le stimulus est composé :

- d'une large cible bleue pouvant apparaitre dans l'un des 4 coins de l'écran;
- d'un nom de commande à sélectionner affiché en haut de l'écran (Figure 13).

En plaçant chaque cible dans un coin différent de l'écran, nous testons les 4 représentations du Leaf menu. Cette configuration a l'avantage de placer l'expérience dans le cas le plus défavorable car les utilisateurs sont confrontés successivement aux différentes symétries horizontales et verticales. Ainsi, les résultats obtenus au cours de cette expérimentation doivent être considérés comme une borne inférieure et seraient certainement supérieurs dans un contexte réel.

Tâche et procédure

Après avoir expliqué le fonctionnement du Leaf menu, les participants s'entraînent à sélectionner des items en mode novice et expert pendant 5 minutes avant de commencer l'expérience. La tâche consiste à appliquer la bonne commande à la cible affichée, aussi vite et précisément que possible. Aucune consigne n'indique le mode de fonctionnement (novice, expert) à utiliser.

Participants et matériel

8 étudiants ont été recrutés au sein de l'Université. Ce sont des bénévoles qui ont reçu une poignée de bonbons pour leur participation. Ils ont réalisé l'expérience sur un PDA-Phone HTC p3600 en position assise et tenant le dispositif d'une seule main, sans s'aider d'un support. Ceci signifie que les participants ont du interagir avec le pouce (sans stylet). Ils n'avaient jamais utilisé ce type de dispositifs mobiles à titre personnel et n'étaient pas familiarisés avec les interfaces gestuelles.

Design

Un essai correspond à l'activation d'un des 7 items du menu sur l'une des 4 cibles. Un bloc comprend $7 \times 4 = 28$ essais afin que chaque item puisse être sélectionné sur chaque cible aux quatre coins de l'écran. Pour cette évaluation, nous avons considéré 10 blocs. Une pause était possible à la fin de chaque bloc. L'ordre d'apparition des items est contrebalancé par un carré latin et la position de la cible est choisie aléatoirement. Chaque participant a réalisé l'expérimentation en une session qui a duré environ 20 minutes. En résumé le design est le suivant :

8 participants *
 1 technique (Leaf menu) *
 4 cibles *
 7 items *
 10 blocs
 = 2240 sélections

1.5.2 Résultats et discussions

La Figure 14 montre l'évolution des 4 types de sélection (réussie en mode novice, ratée en mode novice, réussie en mode expert et ratée en mode expert) en fonction de l'indice du bloc. Une analyse de la variance montre un effet significatif du nombre de sélection réussie en mode expert en fonction de l'indice du bloc ($F_{9,63} = 25.9$, $p < 0.0001$). Ceci souligne que les utilisateurs apprennent les raccourcis gestuels et les exécutent correctement.

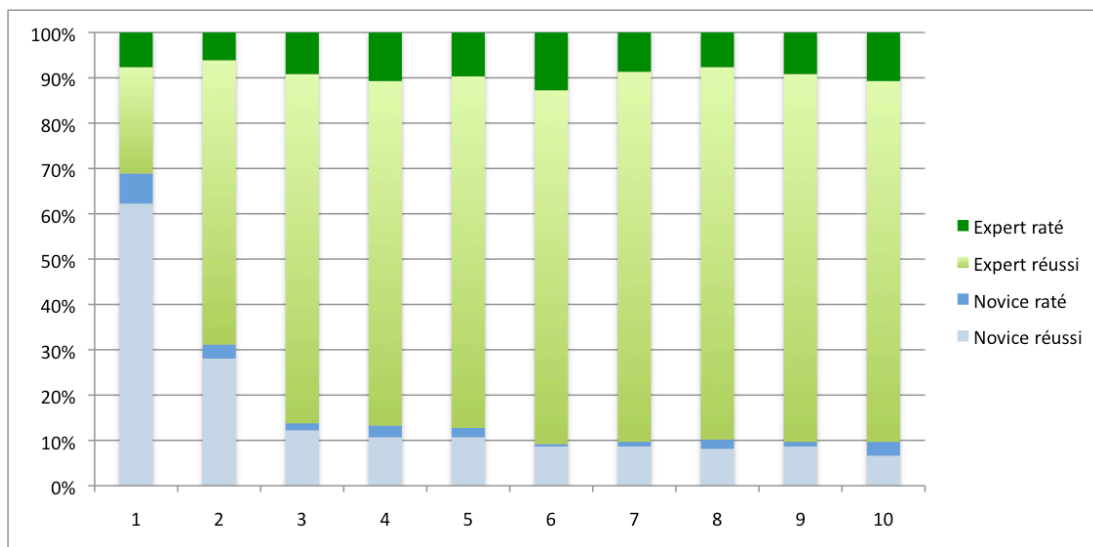


Figure 14 : pourcentage de sélection, réussie en mode novice, ratée en mode novice, réussie en mode expert et ratée en mode expert en fonction de l'indice du bloc.

La Figure 15 détaille l'évolution des sélections effectuées correctement en mode expert en fonction des différents items. Les différents items ne sont pas appris de la même manière ($F_{6,42} = 3.76$, $p < 0.005$). Les trois gestes droits (1, 4, 7) de la Figure 15 sont significativement mieux appris que les deux diagonales courbées (3, 5).

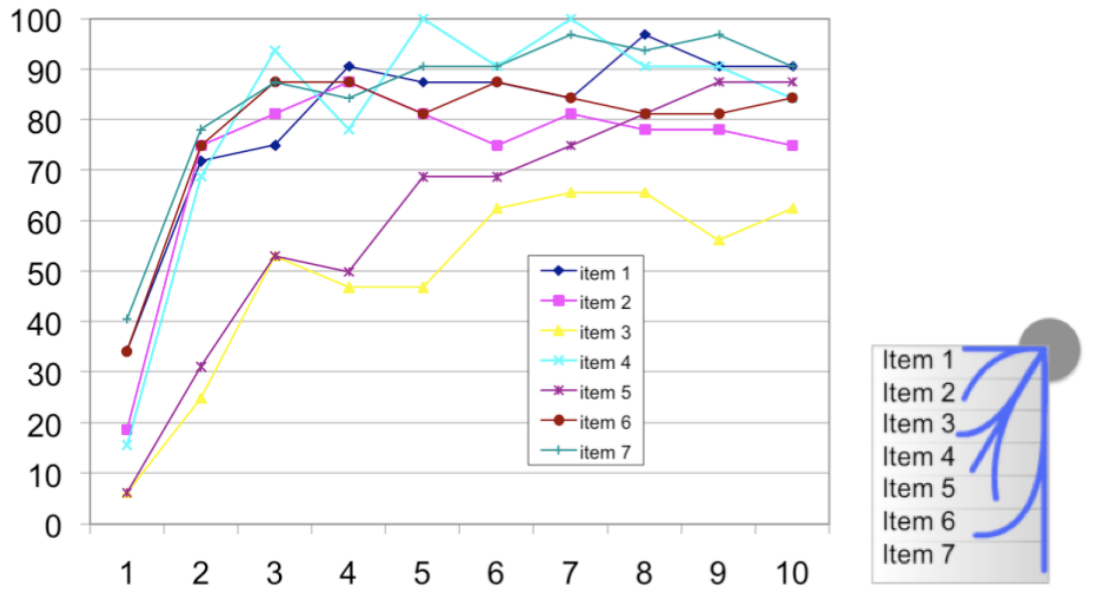


Figure 15 : Pourcentage de sélections correctes en mode expert (SCE) en fonction des différents items et de l'indice du bloc.

Une analyse du premier bloc (7 commandes * 4 positions) montre que les utilisateurs ont été capables de faire plus de 24% de sélections correctes en mode expert pour les items 1, 6, 7 et plus de 15% pour les commandes 2 et 4. Ceci indique d'une part que ces 5 commandes peuvent être apprises rapidement (<4 répétitions) et d'autre part qu'il n'est pas nécessaire d'apprendre indépendamment les 4 gestes de chaque commande car la symétrie du Leaf menu aide à les déduire aisément. Ceci a été confirmé par les entretiens avec les utilisateurs qui n'ont pas été gênés ni par les configurations inversés, ni par la symétrie des gestes correspondants aux items 1, 2, 4, 6, 7. Seul, les gestes diagonaux courbés (item 3 et 5) ont été perçus comme plus difficiles à assimiler. Comme pour le chapitre précédent, ceci montre que la répétition n'est pas le seul facteur favorisant l'apprentissage du mode expert. La simplicité des gestes et leur représentation dans le menu sont des éléments essentiels pour le critère *apprentissage et mémorisation* (section 2 du chapitre 4).

Les utilisateurs ont également souligné qu'une fois les raccourcis gestuels connus (bloc >5), le mode expert (89,1%) était plus précis que le mode novice (84,4%) à cause de la petite taille des items du menu. Ainsi, les utilisateurs utilisaient le mode expert non seulement pour aller plus vite (0,7s en mode expert vs. 3,3s en mode novice), mais aussi pour produire moins d'erreurs. De plus, les résultats en mode expert peuvent être nettement améliorés en révisant l'algorithme de reconnaissance de gestes et en prenant en compte les caractéristiques de l'interaction au pouce. Par exemple, sur les 186 sélections l'item 3 en mode expert, 52 ont été reconnues comme erronées (28%). Mais une analyse des tracés a montrée que 12% des erreurs auraient dû être reconnus corrects mais la morphologie du pouce introduit une variation angulaire qu'il convient d'étudier plus en détail.

Après l'expérience, nous avons collecté les remarques des participants. Ces derniers ont tous affirmé ne pas avoir été gênés par l'effet miroir du Leaf menu. Certains ont même ajouté que cette représentation semblait plus "logique". Ils ont trouvé "amusant" le fait de pouvoir dessiner des marques, même s'ils étaient parfois frustrés que le système ne reconnaisse pas le geste exécuté en particulier sur les diagonales courbées. Ceci confirme qu'il convient d'améliorer notre algorithme de reconnaissance.



Nous retenons que le Leaf menu permet d'apprendre les gestes proposés et de mémoriser leur association aux commandes après peu de répétitions. Nous retenons également que l'effet miroir a été bien accueilli par les participants.

1.6 Synthèse

Dans cette section, nous avons décrit le Leaf Menu, un menu conçu pour les petits écrans tactiles. Le Leaf menu est une nouvelle technique de menus qui combine une représentation linéaire avec un mode expert basé sur l'interaction gestuelle. Un aspect intéressant du Leaf menu est l'introduction d'un *effet miroir* qui inverse l'ordre d'affichage des items lorsque le menu est activé en bas de l'écran. Cet effet miroir a l'avantage de favoriser la sélection des items favoris quelque soit l'endroit où est affiché le menu. De plus, grâce à la redondance des raccourcis gestuels, il est possible de sélectionner les items en mode expert près des bords de l'écran. Enfin, sur ce type de dispositifs, le mode expert des Leaf menu a l'avantage de répondre au problème de l'imprécision de la sélection au pouce en novice grâce à l'interaction gestuelle.

Les résultats encourageants obtenus lors de notre évaluation préliminaire ont montré que les utilisateurs n'étaient pas gênés par l'*effet miroir* et étaient capables de mémoriser les différents gestes associés à une commande après très peu de répétitions. Cette expérimentation suggère également que les gestes du Leaf menu peuvent être appris après peu de répétitions. Ceci avait déjà été observé avec l'expérimentation des Flower menus (chapitre 7), qui soulignait que la répétition n'était pas le seul facteur favorisant l'apprentissage du mode expert. La simplicité des gestes et leur représentation dans le menu sont également des facteurs essentiels pour le critère *apprentissage et mémorisation*.

2 Multi-Touch Menu : un menu pour surfaces multi-points

Nous considérons maintenant le cas des grandes surfaces interactives multi-points. Nous présentons le Multi-Touch Menu (MTM), une nouvelle technique de menus qui exploite tous les doigts de la main. Nous motivons d'abord nos travaux par la présentation du modèle de la main de Lin [Lin 00] comportant 21 degrés de liberté. Parmi les rares dispositifs tirant parti de ces degrés de liberté, nous nous focalisons sur les surfaces multi-points. Nous en décrivons les principaux travaux en relation avec le Multi-Touch Menu. Enfin, nous exposons le principe et les propriétés interactionnelles du Multi-Touch Menu.

2.1 Degrés de liberté de la main

La main est constituée de quatre doigts longs (l'index, le majeur, l'annulaire et l'auriculaire) du pouce, de la paume et du talon de la main (entre le poignet et la paume) (Figure 16). C'est un organe important chez l'être humain qui dispose d'une palette d'actions très large (saisir, manipuler, pointer, etc.). Nous présentons le modèle de la main de Lin [Lin 00] qui définit 21 degrés de liberté.

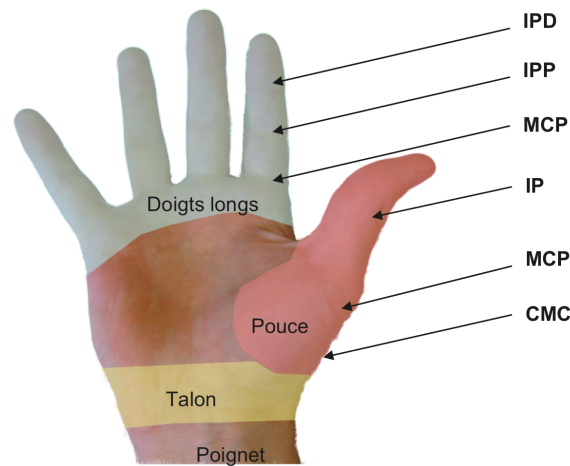


Figure 16 : Différentes parties de la main et les nom des principales articulations.

Articulations

Le poignet permet les mouvements globaux de la main par rapport à l'avant bras. Il offre 6 degrés de liberté : 3 translations et 3 rotations.

Les doigts longs sont constitués de 3 phalanges et 3 articulations (Figure 17) :

- l'articulation métacarpo-phalangienne (**MCP**) à la base du doigt possède 2 degrés de liberté : flexion-extension et adduction-abduction illustrés Figure 18 ;
- l'articulation inter-phalangienne proximale (**IPP**) possède 1 degré de liberté en flexion-extension ;
- l'articulation inter-phalangienne distale (**IPD**) possède également 1 degré de liberté en flexion-extension.

Les doigts longs se partagent donc $4*(2+1+1) = 16$ degrés de liberté.

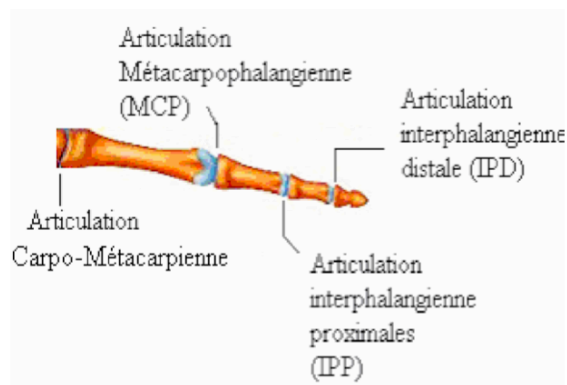


Figure 17 : 3 phalanges d'un doigt long. Illustration extraite de [Ouhaddi 98].

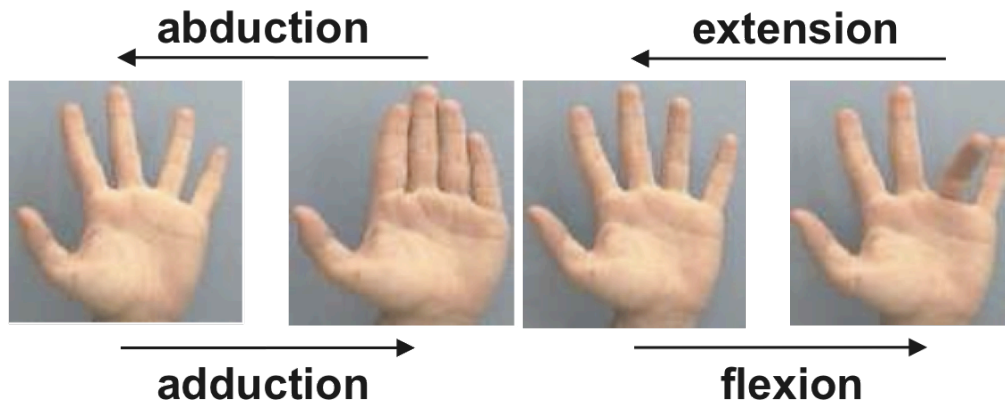


Figure 18 : A gauche, les mouvements d'adduction et d'abduction. A droite, les mouvements d'extensions et de flexion. Illustration extraite de [Ouhaddi 98].

Le pouce a la particularité d'être opposé aux doigts longs ce qui permet la saisie d'objets. Il possède également trois articulations mais qui sont différentes de celles des doigts longs.

- l'articulation carpo-métacarpe (**CMC**), à la base du pouce, possède 2 degrés de liberté ;
- l'articulation métacarpo-phalangienne (**MCP**) possède comme les doigts longs 2 degrés de liberté : flexion-extension et adduction-abduction ;
- l'articulation interphalangienne (**IP**) possède 1 degré de liberté en flexion-extension.

Le pouce a donc $2+2+1 = 5$ degrés de liberté.

En résumé, Lin [Lin 00] propose un modèle des mouvements de la main avec 27 degrés de liberté (6 pour le poignet, 16 pour les doigts longs et 5 pour le pouce). Cependant, certains mouvements des doigts sont parfois impossibles ou contraints par les mouvements des autres doigts.

Contraintes

Les contraintes peuvent être organisées selon deux types [Lin 00] : les *contraintes statiques* et les *contraintes dynamiques*.

- Les contraintes statiques réfèrent aux amplitudes limites des mouvements. Par exemple, l'angle d'abduction-adduction (Figure 18) de l'index et de l'auriculaire est compris entre -15° et $+15^\circ$.
- Les contraintes dynamiques sont issues des limites imposées aux articulations pendant les mouvements des doigts. Par exemple, lorsque le majeur bouge, l'annulaire est naturellement entraîné avec lui [Häger 00]. Ces deux doigts sont très dépendants. A l'opposé, l'index et le pouce sont indépendants.

Finalement, Lin réduit son modèle à 21 degrés de liberté sans dégradation majeure des performances [Lin 00]. Pourtant nous observons que la plupart des dispositifs, comme la souris, exploitent faiblement les possibilités de la main car ils utilisent relativement peu les degrés de libertés des doigts : une flexion de l'index ou du majeur permet d'appuyer sur les boutons, mais les autres doigts sont restreints au maintien de la souris. Des dispositifs d'interaction comme le Z-Glove ou le DataGlove (Figure 19), ont été conçus pour exploiter les différents degrés de liberté de la main [Zimmerman 87]. Des systèmes de commandes ont été développés pour ce type de dispositifs dont nous trouvons une taxonomie dans [Dachselt 07] (présentée à la section 1 du chapitre 2). Dans cette section, nous nous concentrons sur les surfaces interactives multi-points pour lesquelles nous avons conçu notre technique de menus, le Multi-Touch Menu.



Figure 19 : DataGlove. Illustration extraite de <http://www.virtualtechnologiesltd.com/>.

2.2 Surfaces interactives multi-points

Les surfaces interactives multi-points offrent de nouvelles opportunités pour augmenter la bande passante interactionnelle entre l'homme et la machine. Non seulement, l'utilisateur peut interagir avec plusieurs doigts, mais il peut parfois aussi interagir avec d'autres parties de la main comme la paume, ou le côté [Wu 03]. Ces nouvelles possibilités ont suscité de nombreuses études [Wellner 93, Matsushita 97, Dietz 01, Han 05, JazzMutant, Hodges 07] depuis le DigitalDesk (Figure 20), en 1993 [Wellner 93]. Cet essor pour l'interaction multi-points est à l'origine du groupe de travail TABLETOP depuis 2006 ainsi que des produits commerciaux comme l'iPhone [iPhone] et la « Microsoft Surface » [MicrosoftSurf]. En France, trois industriels proposent depuis peu des surfaces multi-points :

- Immersion (<http://www.immersion.com/>)
- Intuilab (<http://www.intuilab.com/>)
- JazzMutant(<http://www.jazzmutant.com/>)

Parmi cette grande variété de technologies, dont nous trouvons une revue dans [Hodges 07], nous avons développé une version de notre menu Multi-Touch sur deux technologies que nous présentons ci-après.

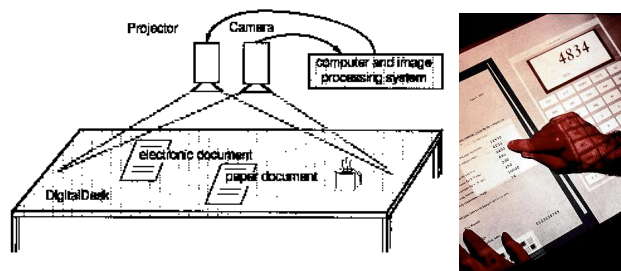


Figure 20 : DigitalDesk. Illustration extraite de [Wellner 93].

2.2.1 Deux technologies multi-points

Technologie Jeff Han [Han 05]

La technologie Jeff Han [Han 05] utilise un vidéoprojecteur et une caméra placés derrière un écran translucide. Ceci permet d'éviter des problèmes d'occultation présents lorsque qu'une partie du corps peut se trouver entre la caméra et les doigts [Wellen 93]. La technologie de Jeff Han repose sur la réflexion totale contrariée (« Frustrated Total Internal Reflection » : FTIR), illustrée Figure 21. Sur les bords d'un écran tactile FTIR, une rangée de LED émet un faisceau lumineux infrarouge dans l'épaisseur de la dalle en verre de l'écran. Lorsque l'on y pose un doigt, le phénomène de réflexion totale est brisé localement car le doigt reflète la lumière infrarouge différemment de l'air ambiant. Cette perturbation est alors détectée précisément par la caméra.

Cette technologie a de nombreux avantages : elle est facile à mettre en œuvre, peu chère et facilement adaptable à des écrans de très grande taille.

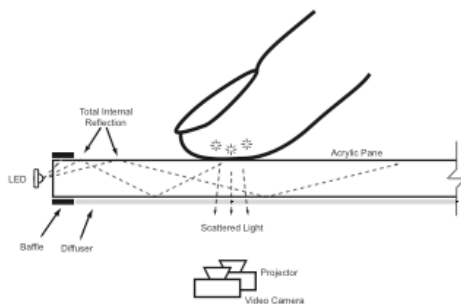


Figure 21 : Technologie de Jeff Han [Han 05] basé sur la réflexion totale contrariée (FTIR). Illustration extraite de [Han 05].

Stantum [JazzMutant]

Le Stantum [JazzMutant] est l'un des seuls écrans tactiles multi-points proposés dans le commerce (Figure 22) d'un prix relativement élevé. Il reprend la technologie du Lemur, une table de mixage virtuelle apparue en 2004. Le Stantum est un écran LCD 15.4 pouces avec des capteurs résistifs propriétaires. Ces capteurs permettent d'obtenir une dizaine de points de contact simultanément. Il a cependant plusieurs inconvénients :

- la résolution en entrée est faible (128x120) ;
- aucune information sur la pression n'est fournie ;
- une force supérieure à 20N doit être appliquée pour que les points puissent être détectés.

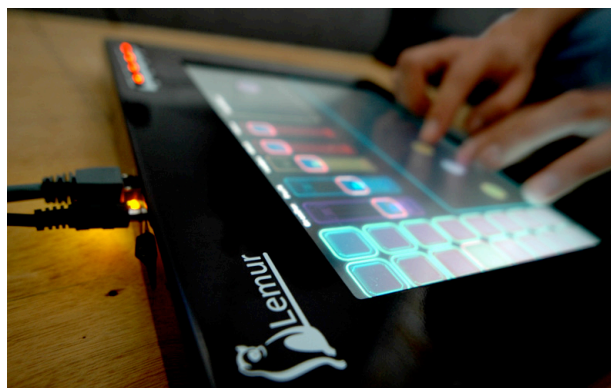


Figure 22 : Lemur : écran LCD tactile permettant de capturer une dizaine de points de contact simultanément. Illustration extraite de [JazzMutant].

2.2.2 Exemples de techniques d'interaction multi-points

Comme nous l'avons dit précédemment, les surfaces multi-points suscitent actuellement de nombreux travaux. Les innovations technologiques s'accompagnent d'études sur les techniques d'interaction. A titre d'exemples, nous présentons le vocabulaire de gestes conçu par Apple pour l'iPhone ainsi qu'une autre technique [Wu 03] qui exploite d'autres parties de la main que les doigts. Nous nous concentrons ensuite sur les techniques de menus multi-points existantes.

Vocabulaire de gestes avec les doigts d'Apple

Apple a proposé un ensemble de gestes simples utilisant plusieurs doigts pour les Touchpad de leurs ordinateurs portables (Figure 23) et l'iPhone. La technique d'interaction multi-points la plus connue est certainement le « Pincer & écarter » (Pitch & Expand) qui sert à

zoomer un élément (Figure 23-a). La distance entre les doigts définit le facteur de zoom. La rotation d'un objet se fait par une rotation avec deux doigts (« Rotate » Figure 23-b). Le « défilement » (scroll) horizontal ou vertical permet de faire défiler un document en réalisant une translation de deux doigts rapprochés (Figure 23-c). Le « Swipe » consiste à effectuer une translation rapide avec un doigt. Ce geste a été utilisé dans l'iPhone pour faire défiler une liste et a été étendu à trois doigts (« Three-finger Swipe », Figure 23-d) pour faire défiler rapidement les photos. Un « double tap », c'est-à-dire un appui simultané de deux doigts rapprochés, permet d'ouvrir un menu contextuel (Figure 23-e). Enfin, Apple a breveté d'autres couples (geste/fonctionnalité) dans [PatentApple 07]. Par exemple, trois doigts qui se rapprochent permettent de « couper » un élément. Pour « coller », l'utilisateur écarte les trois doigts (Figure 24).

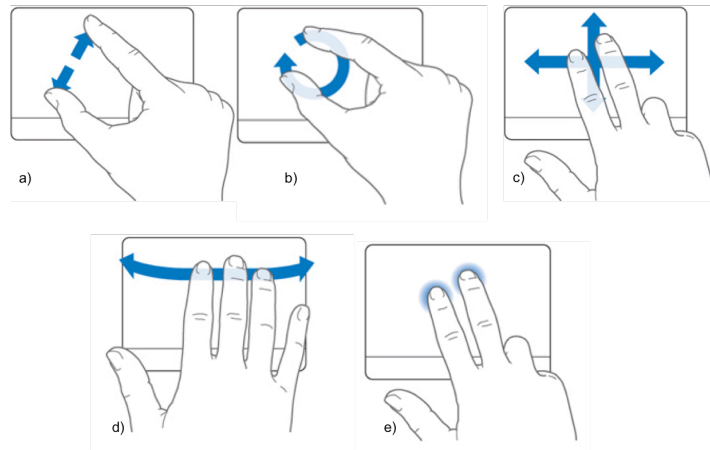


Figure 23 : vocabulaire gestuel d'Apple. a) le « Pitch & Expand » utilisé pour zoomer un élément. b) le « Rotate » pour la rotation d'un objet. c) Le « Scroll », translation à deux doigts pour faire défiler un document. d) le « Three-finger Swipe » utilisé pour « jeter » un document. e) le « Double tap », appui simultané de deux doigts pour ouvrir un menu. Illustration extraite de [TrackpadGestures].

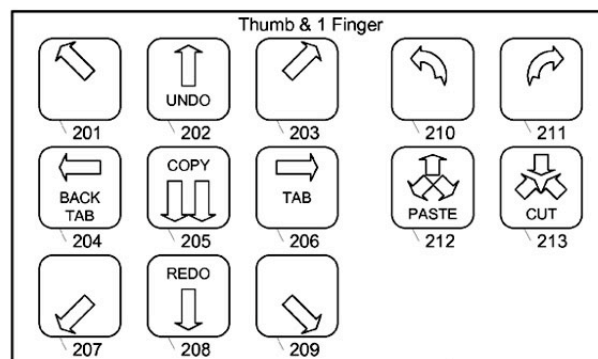


Figure 24 : Extrait du brevet « Mutli-Touch Gesture Dictionary ». Illustration extraite de [PatentApple 07].

Vocabulaire de gestes avec le revers de la main [Wu 03]

Wu propose dans [Wu 03] un ensemble de techniques d'interaction exploitant des parties de la main rarement utilisées. Le « Sweeping vertical » (Figure 25) consiste par exemple à exécuter un geste d'un revers de main pour déplacer un groupe d'objets.



Figure 25 : « sweeping vertical » pour déplacer un groupe d'objets. Illustration extraite de [Wu 03].

Il propose également une technique avec les deux mains verticales (« Two vertical Hands ») pour délimiter une zone de sélection (Figure 26-gauche). Lorsque les mains sont en coin (« Two Corner-Shaped Hands »), l'utilisateur définit une zone privée visible que de l'utilisateur (Figure 26-droite).

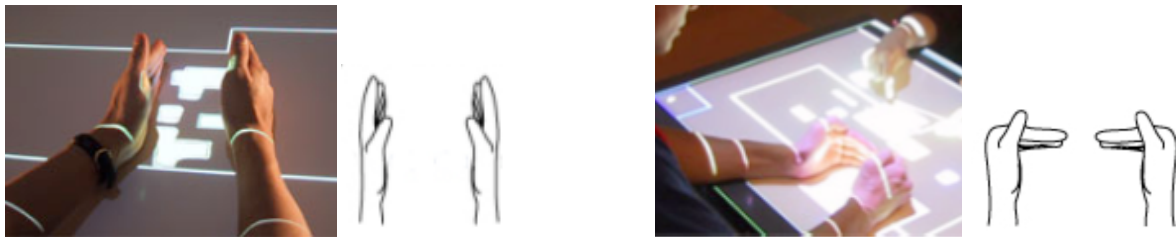


Figure 26 : A gauche, les côtés de la main délimitent une zone de sélection. A droite, les mains positionnées en coin définissent une zone privée visible que de l'utilisateur. Illustrations extraites de [Wu 03].

2.3 Techniques de menus multi-points

Dual Finger X-Menu

Le « Dual Finger X-Menu » [Benko 06], illustré à la Figure 27, est un menu circulaire permettant de sélectionner de petites cibles sur écran tactile. Le menu est activé lorsque deux doigts sont en contact sur la table. Les différentes options du menu permettent de zoomer une partie de l'interface ou de modifier la vitesse du curseur pour effectuer des sélections précises.



Figure 27 : « Dual Finger X-Menu [Benko 06]. Le menu circulaire est activé dès que deux doigts sont posés sur la table. Illustration extraite de [Benko 06].

Menu de Brandl

Le menu de Brandl [Brandl 08], illustré à la Figure 28, est un menu semi-circulaire qui exploite deux doigts. L'utilisateur active le menu en posant un doigt sur la table et en attendant un certain délai. En utilisant le majeur pour l'activation, l'utilisateur peut alors facilement sélectionner une commande avec son index.

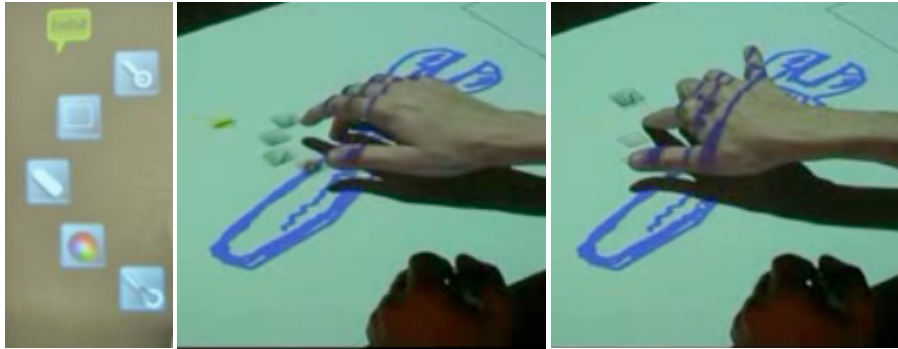


Figure 28 : menu de Brandl. Un menu semi-circulaire qui s'active de préférence avec le majeur permettant à l'index de sélectionner un item. Illustration extraite de [Brandl 08].

FurniturePalette

La « FurniturePalette » [Wu 03], illustrée à la Figure 29, est un menu circulaire que l'on active en tapant deux fois avec le pouce. Cette solution compense l'absence du bouton droit de la souris. La sélection d'un item avec le pouce fait apparaître une toolglass [Bier 93] de 4 éléments. Ces éléments peuvent être activés avec l'index. La « Furniture Palette » utilise donc le pouce au premier niveau de la hiérarchie et l'index au second niveau.

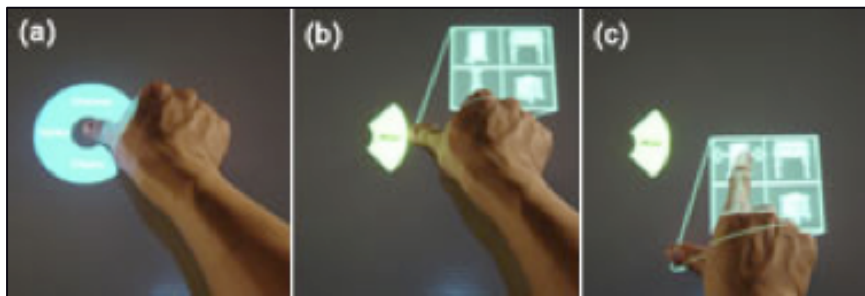


Figure 29 : La « Furniture Palette » [Wu 03]. L'utilisateur « double tap » avec le pouce pour activer le menu. La sélection d'un item donne accès à une Toolglass qui est manipulée avec l'index. La technique implique donc l'utilisation de deux doigts, le pouce au premier niveau de la hiérarchie et l'index au second niveau. Illustration extraite de [Wu 03].

Le menu de Gäser

Le menu de Gäser [Gläser 07] (dont la seule référence dont nous disposons est une vidéo sur Youtube), est un menu semi-circulaire qui est activé en posant toute la main à plat sur la table. Une orientation de celle-ci permet de sélectionner un item. Pour activer celui-ci, soit l'utilisateur attend la fin du délai, soit il « tap » avec un doigt de l'autre main. Le menu disparaît une fois que l'utilisateur a retiré la main de la table.

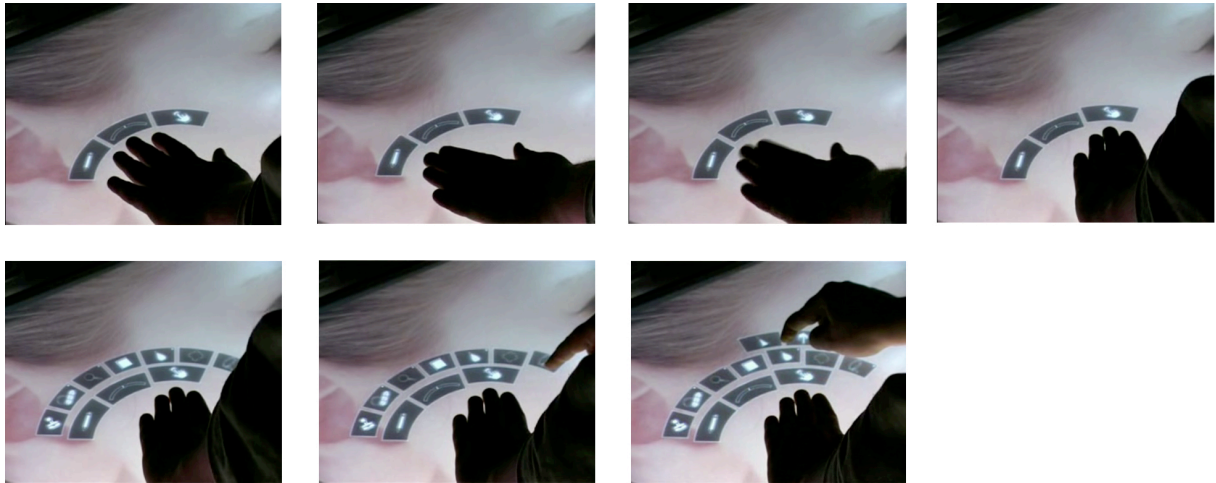


Figure 30 : menu de Gäser : le menu est ouvert tant que toute la paume est en contact avec la table. L'orientation de la paume permet de sélectionner un élément. Il est activé après un délai. Pour activer les éléments plus profonds, il est nécessaire d'utiliser une autre main. Illustration extraite de [Gläser 07].

Menu de Koike

Le menu de Koike [Koike 02], illustré Figure 31, est également un menu semi-circulaire hiérarchique. Pour l'activer, l'utilisateur doit poser ses deux mains sur la table et replier tous les doigts de la main gauche. Il sélectionne un item au premier niveau avec l'index en faisant un mouvement latéral du poignet. Si l'item est hiérarchique, le sous-menu est superposé et centré autour de l'index.

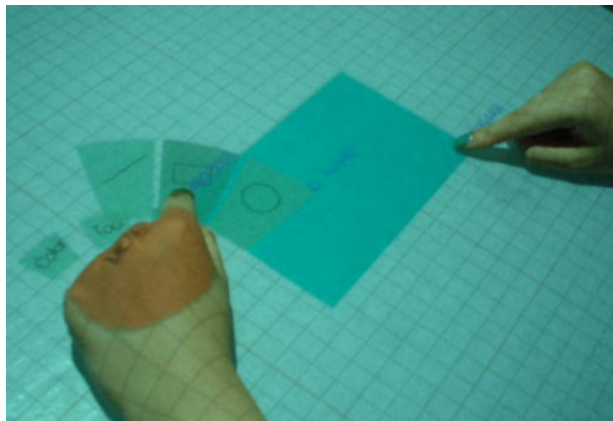


Figure 31 : menu de Koike [Koike 02]. Pour activer le menu, l'utilisateur doit poser les deux mains sur l'écran et replier tous les doigts de la main gauche.

En résumé, les trois premiers menus, le X-menu [Benko 06], le menu de Brandl [Brandl 08] et la « FurniturePalette » [Wu 03] contiennent un nombre limité de commandes et exploitent faiblement les degrés de liberté de la main car ils n'utilisent que deux doigts. De plus, l'activation avec deux doigts, ou l'activation avec un double tap du pouce peuvent ne pas être compatibles avec d'autres techniques d'interaction. Au contraire, l'activation avec le plat de la main [Gäser] ou le repliement des doigts [Koike 02] est suffisamment spécifique pour ne pas créer d'ambiguïtés avec d'autres techniques d'interaction mais ne permet d'utiliser pleinement les doigts. Néanmoins, L'empreinte caractéristique laissée par ces deux techniques a l'avantage de permettre d'orienter les items afin qu'ils soient lisibles facilement par l'utilisateur. Ils permettent de contenir davantage de commandes que les menus précédents car ils sont hiérarchiques. Enfin, toutes ses techniques exploitent peu les cinq doigts de la main et ne proposent pas une navigation efficace dans le système de menus (à cause de l'absence de prévisualisation).



Nous retenons que malgré les nombreux travaux dédiés à l'interaction sur des surfaces multi-points, il existe peu de menus exploitant plusieurs doigts.

2.4 Multi-Touch Menu (MTM)

Nous proposons maintenant une nouvelle technique de menus appelée Multi-Touch Menu (MTM) dont l'originalité est d'utiliser tous les doigts de la main pour la sélection de commandes contrairement aux menus ci-dessus. Nous décrivons d'abord le principe de fonctionnement du Multi-Touch Menu en distinguant l'activation du menu de la sélection d'un item. Nous détaillons ensuite ses propriétés interactionnelles au regard de notre espace MenUA.

2.4.1 Activation du Multi-Touch Menu

Sur les surfaces multi-points, l'utilisateur interagit directement avec ses doigts et ne dispose donc pas du bouton droit de la souris pour activer le menu. Plusieurs types d'*activateur* ont été proposés qui ont chacun des inconvénients :

- l'usage d'un délai ralentit l'interaction ;
- le « double tap » utilisé dans [Wu 03] n'est pas toujours possible car il est souvent utilisé pour activer un objet du domaine ;
- la pose du plat de la main [Gäser 07] ou le repliement des doigts [Koike 02] ne permettent pas de laisser les doigts libres pour interagir.

Nous proposons ici deux types d'*activateurs* pour le Multi-Touch Menu : le premier repose sur *la pression du talon de la main*, l'autre sur *la pose simultanée des cinq doigts*.

Talon de la main

Cet *activateur* consiste à poser le talon de la main (section 2.1) pour ouvrir le menu. Cette solution utilise une partie de la main jusqu'ici inexploité et qui a plusieurs avantages :

- **Compatibilité interactionnelle.** Cet activateur peut facilement être intégré à une application sans introduire de conflits avec les techniques d'interaction existantes. En effet, la trace (l'empreinte) laissée par le talon de la main est très spécifique et habituellement inutilisée.
- **Adaptabilité à l'utilisateur.** La forme spécifique du talon de la main permet à la fois de déterminer la main utilisée ainsi que son orientation. Le menu peut donc s'afficher en adéquation avec la position de l'utilisateur autour de la table.
- **Doigts libres.** En utilisant le talon de la main, tous les doigts sont libres pour d'autres techniques. Ceci permet par exemple d'activer le menu et de sélectionner une commande simultanément. Nous détaillons ce point dans la section 2.5.1.

Cet activateur a été développé sur une table de type Jeff Han [Han 05] (section 2.2). Les événements reçus, des ellipses (position, largeur, hauteur), ne permettent pas d'avoir l'empreinte exacte du talon sur la table. Celle-ci a été simulée par une ellipse avec une aire comprise entre 2.5cm² et 5cm². L'orientation de la main est obtenue par la dynamique du déplacement de l'ellipse. Une perspective consisterait à réaliser nous-mêmes le traitement de l'image afin de pouvoir identifier plus finement l'empreinte du talon de la main.

Des tests informels montrent que cet activateur peut se révéler fatigant pour le poignet lorsqu'il est utilisé de manière intensive dans certaines positions. Les Figure 32-b et -c montrent en effet que lorsque l'utilisateur est debout devant une table (surtout si celle-ci est inclinée), l'angle entre la main et l'avant bras est « fermé » et peut provoquer rapidement de la fatigue. Pour cette raison, nous proposons un autre type d'activateur mieux adapté à la position debout.

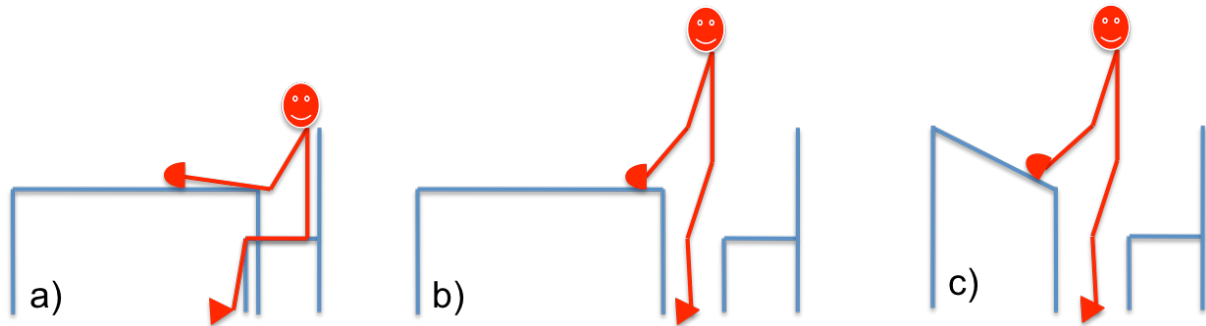


Figure 32 : manipulation du MTM, a) l'utilisateur assis à une table horizontale, b) l'utilisateur debout à une table horizontale, c) debout à une table inclinée. Dans les cas b) et c) l'angle entre la main et l'avant bras est fermé et peut provoquer de la fatigue au niveau du poignet.

Cinq doigts simultanés

Cet activateur consiste à poser simultanément les cinq doigts sur la table pour ouvrir le menu. Il reste compatible avec les autres styles d'interaction car habituellement il est rare de poser simultanément cinq points de contacts dans une zone délimitée (10cm x 5 cm) avec une répartition compatible avec un espacement normal entre les doigts.

Cet activateur permet également de déterminer la main utilisée et son orientation grâce à la répartition spécifique des doigts de la main comme le montre la Figure 33. Il fournit un référentiel précis pour positionner les différents éléments graphiques du menu comme les boutons. Les cinq points de contact donnent en effet les positions clés des doigts pour disposer au mieux les différents items et s'adapter aux différentes tailles de main.

Cet activateur n'est pas fatigant pour le poignet car il ne nécessite pas de mouvements particuliers de celui-ci, même lorsque l'utilisateur est debout. Cependant, il a l'inconvénient de ne plus laisser les doigts libres : l'utilisateur ne pourra pas activer le menu et sélectionner une commande en même temps car ces actions utilisent les doigts. L'activation du menu et la sélection de commandes ne peuvent pas être réalisées simultanément.

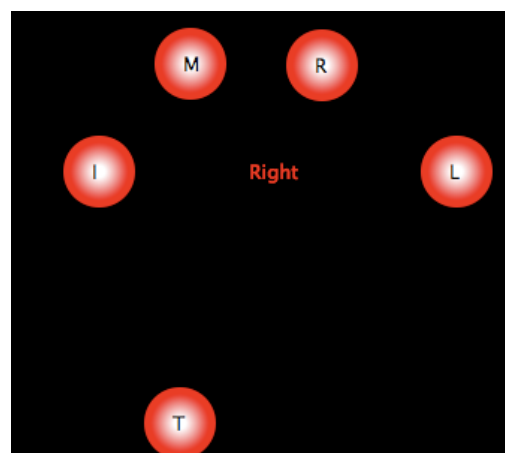


Figure 33 : Résultats de notre algorithme des cinq doigts sur un écran Stantum. Chaque point de contact a été affecté correctement à un doigt (T : thumb, I : index, M : middle, R : ring, L : little) et la main droite (Right) a été reconnue).

Cet activateur a été implémenté sur un écran Stantum [JazzMutant] (section 2.2) capable de reconnaître une dizaine de points de contact simultanément. Le menu est activé lorsque 5 événements « Down » sont 1) détectés dans une fenêtre temporelle de 100ms, 2) regroupés dans un cercle de 270 px (10cm) de rayon et 3) n'appartenant pas à un autre « Chord » (geste multi-points). Nous utilisons ensuite un algorithme pour associer à chaque point de contact le doigt correspondant. Cet algorithme, illustré à la Figure 34, repose sur l'hypothèse vraisemblable que « le pouce est le doigt dont la distance avec son plus proche voisin est la plus grande. En d'autres termes, les distances séparant un doigt à son plus proche voisin sont relativement petites sauf pour le pouce, pour lequel la distance avec son plus proche voisin, l'index, est importante.

L'algorithme est le suivant :

1. Construire la matrice des distances entre chaque point (une matrice 5*5).
2. Pour chaque point P_i , déterminer le point P_j tel que la distance soit minimale (recherche du plus proche voisin).
3. Pour tous les couples de plus proches voisins ($P_i ; P_j$) obtenus, déterminer le couple qui a la plus grande distance.
4. Le point P_i correspond alors au pouce et le point P_j correspond à l'index
5. De proche en proche, en déduire les autres doigts par distance minimale.

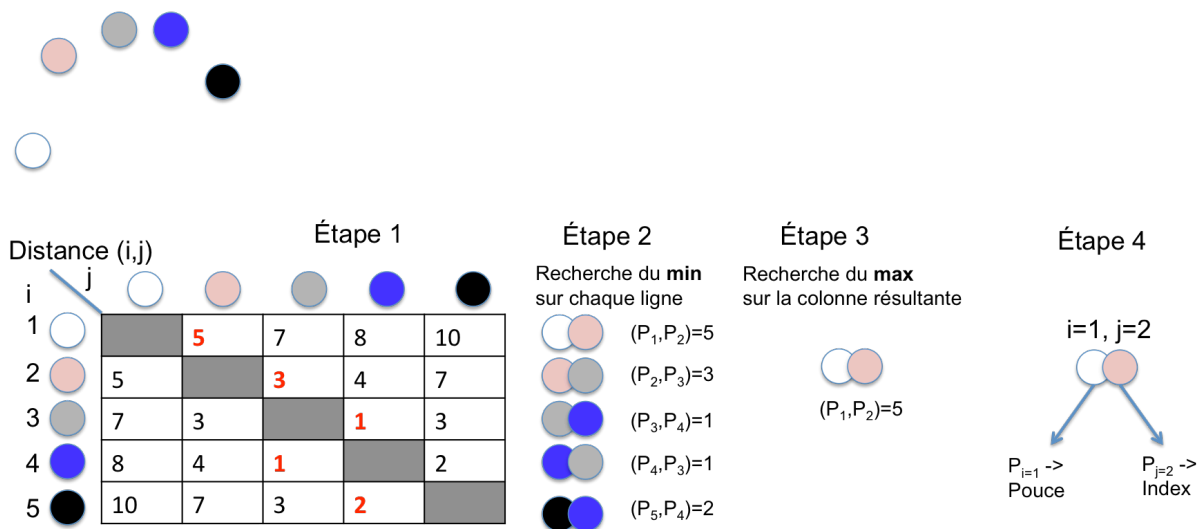


Figure 34 : Algorithme pour identifier le doigt associé à chaque point.

Les premiers essais de cet algorithme sur un écran Stantum sont positifs. L'algorithme atteint ses limites lorsque le pouce est posé très près de l'index ou lorsqu'il est posé plus près du petit doigt que de l'index. Cependant, ces deux situations nécessitent une distorsion du pouce qui ne correspond pas à un geste naturel de la main.



Nous retenons deux types d'activateur pour le MTM (avec le talon de la main ou les cinq doigts) qui sont tous deux compatibles avec les autres styles d'interaction et permettent de déterminer la main utilisée et son orientation.

L'activation avec les cinq doigts réduit la fatigue du poignet et permet de positionner précisément les éléments du menu.

Nous décrivons maintenant le principe de fonctionnement du Multi-Touch Menu.

2.4.2 Sélection d'une commandes avec le MultiTouch Menu

Une fois que l'utilisateur a activé le Multi-Touch Menu (MTM) avec l'un des deux activateurs présentés ci-dessus, le menu apparaît après un délai de 300ms. La représentation graphique du MTM (Figure 35), est décomposée en deux parties :

- Sous le pouce, un disque décomposé en 8 secteurs permet d'afficher le premier niveau de la hiérarchie.
- Sous les doigts longs, 8 boutons de forme elliptique permettent d'afficher le second niveau de la hiérarchie. Ils sont organisés en 4 colonnes, chacune affectée à un doigt long.

Nous avons choisi d'affecter le premier niveau de la hiérarchie au pouce car c'est le doigt qui a le plus de degrés de liberté et qui est le plus indépendant (section 2.1).

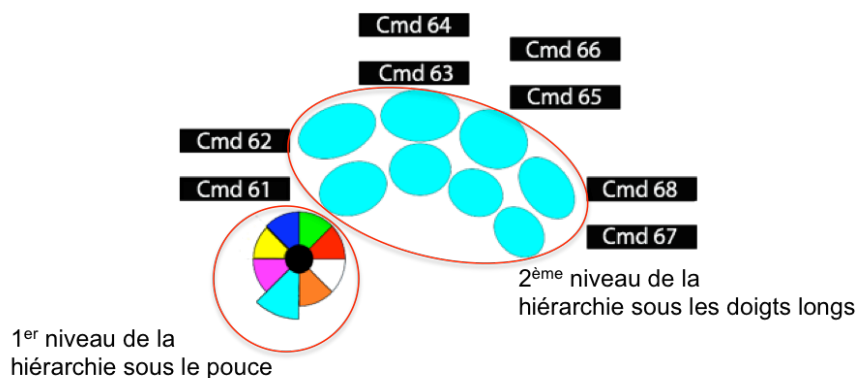


Figure 35 : Deux niveaux de hiérarchie du Multi-Touch Menu.

Pour activer une commande, l'utilisateur sélectionne d'abord un item dans le disque (1^{er} niveau de la hiérarchie) à l'aide du pouce. Il active cet item soit par un « tap » (appui/relâchement), soit par un geste allant du centre du disque vers l'item. Le sous-menu associé à cet item hiérarchique est alors affiché sous les doigts longs. Une conséquence intéressante de cette organisation est qu'un mouvement circulaire du pouce sur le disque permet d'offrir un aperçu rapide des différents sous-menus sous les doigts longs (Figure 36). Pour sélectionner un item du second niveau de la hiérarchie, l'utilisateur utilise le doigt correspondant à la rangée où se situe l'item désiré. Un mouvement d'inflexion (section 2.1) permet de différencier les deux items. Un simple « Tap » de ce doigt permet alors d'activer la commande associée.

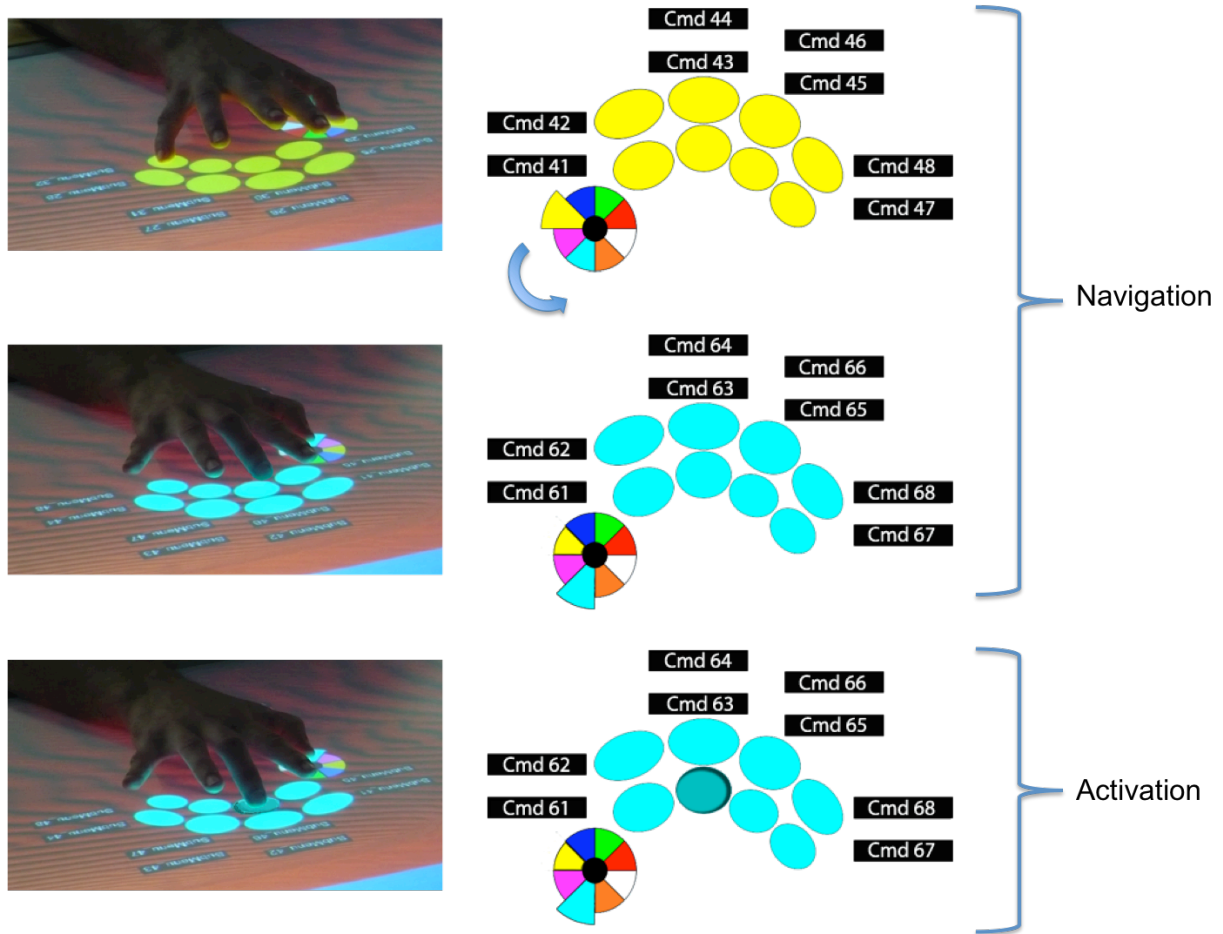


Figure 36 : Multi-Touch Menu. L'utilisateur fait un mouvement circulaire avec le pouce pour explorer les différents sous-menus. Une fois, la commande désirée trouvée, elle est activée avec le doigt long correspondant.

En mode expert, l'utilisateur exécute les mêmes opérations qu'en mode novice, mais le menu n'est plus affiché.



Nous retenons que le Multi-Touch Menu (MTM) est un menu hiérarchique basé sur la technologie multi-points. Il utilise le pouce pour sélectionner un item du premier niveau puis l'un des doigts longs pour le second niveau.

2.5 Propriétés interactionnelles du Multi-Touch Menu

Nous décrivons maintenant les propriétés du Multi-Touch Menu relativement à notre espace MenUA. Nous présentons d'abord les propriétés associées au critère d'utilisabilité, puis celles associées au critère d'applicabilité.

2.5.1 Rapidité et précision

Recherche visuelle

Le Multi-Touch Menu permet la prévisualisation des sous-menus. A l'aide d'un mouvement circulaire du pouce, l'utilisateur peut rapidement inspecter l'ensemble des sous-menus. De plus, ces derniers sont toujours affichés au même endroit, sous les doigts longs, évitant des mouvements oculaires inutiles.

Un inconvénient possible du Multi-Touch Menu est l'occultation du menu par la main (section 2.5.5).

Sélection d'un item

La particularité du MTM est d'utiliser plusieurs doigts. Le pouce sélectionne un item au premier niveau (avec un « tap » ou un « stroke ») tandis qu'un doigt long sélectionne un item au second niveau (avec un « tap »). Comme pour les Multi-Stroke menus, l'utilisateur fait deux gestes simples et indépendants au lieu d'un geste plus compliqué. La différence, est que grâce à la technologie multi-points, l'exécution des gestes n'est plus forcément séquentielle, mais peut être concomitante (léger parallélisme, Figure 37) en mode novice. En mode expert, nous verrons comment permettre des gestes totalement parallèle (Figure 38) en utilisant une fenêtre temporelle.

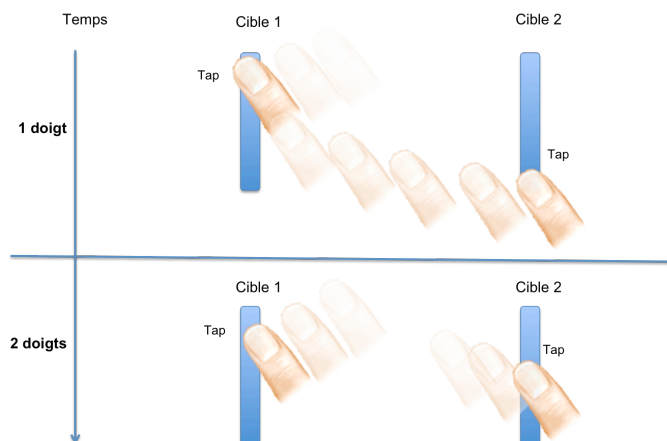


Figure 37 : Pointage séquentiel de deux cibles. En haut avec un seul doigt : l'utilisateur sélectionne les deux cibles séquentiellement. En bas avec deux doigts : l'utilisateur sélectionne les deux cibles de manière concomitante (léger parallélisme).

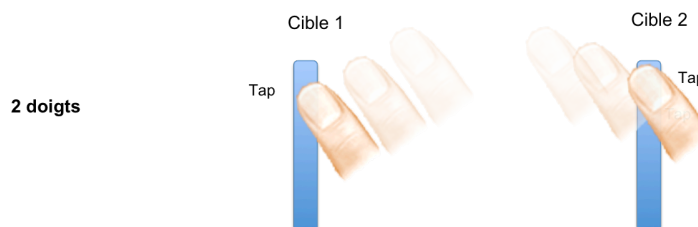


Figure 38 : Pointage de deux cibles en parallèle.

Activation d'un item

L'activation d'un item s'effectue par le relâchement du doigt sur la forme elliptique correspondante. Ces formes sont plus larges lorsque le menu est activé par le talon car il est difficile de prévoir précisément où se trouve l'extrémité des doigts de l'utilisateur. Une solution consiste à calibrer le menu pour chaque utilisateur.

Mode expert

Le mode expert du Multi-Touch Menu permet l'*accès direct* à une commande de profondeur 2 en autorisant des actions simultanées des doigts. Pour cela, nous proposons d'utiliser une fenêtre temporelle de 300ms. Cette fenêtre temporelle évite que l'utilisateur sélectionne accidentellement l'item du second niveau de la hiérarchie avant celui du premier niveau en posant un doigt long avant le pouce. Cette fenêtre offre ainsi une plus grande flexibilité en n'imposant pas un ordre sur les actions de l'utilisateur. Lorsque le délai expire (fin de la fenêtre temporelle), le système analyse les deux points de contacts pour activer la commande choisie par l'utilisateur. Pour cela, le système associe aux deux points de contacts le doigt correspondant à l'aide d'une variante de notre algorithme présenté dans la section 2.4.1. Le sous-menu est déduit en identifiant la trace associée au pouce. L'item final est ensuite activé grâce à la trace laissée par le doigt long.

Avec l'activation du menu par le talon, l'utilisateur peut poser simultanément le talon, le pouce et l'index pour activer une commande comme le montre la Figure 39 : ceci correspond alors à une simple configuration ou posture de la main qui peut être perçue comme un seul mnème pour un utilisateur expérimenté.

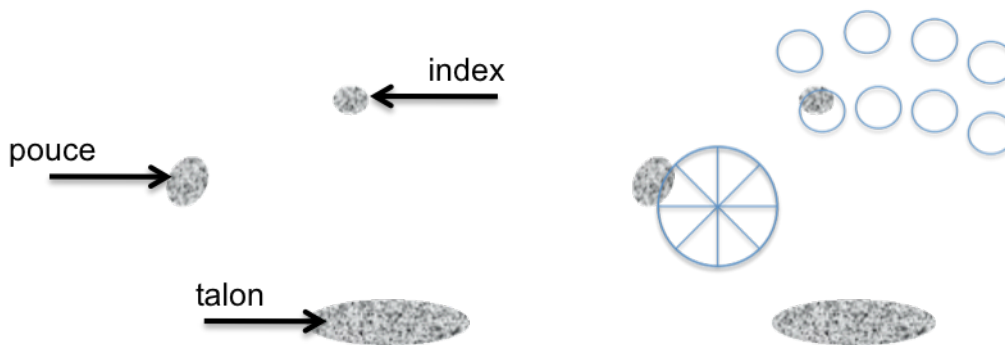


Figure 39 : Activation d'une commande en mode expert avec le talon comme type d'activateur. A gauche, la trace associée à la posture de la main. A droite, son interprétation, en utilisant le talon comme référentiel pour placer les différents éléments du menu.

Avec l'activation par les cinq doigts, l'utilisateur est forcé de réaliser deux actions (Figure 40). D'abord, il tape avec les cinq doigts pour activer le menu et indiquer la position des différents éléments, puis il positionne ses doigts par rapports à ce référentiel.

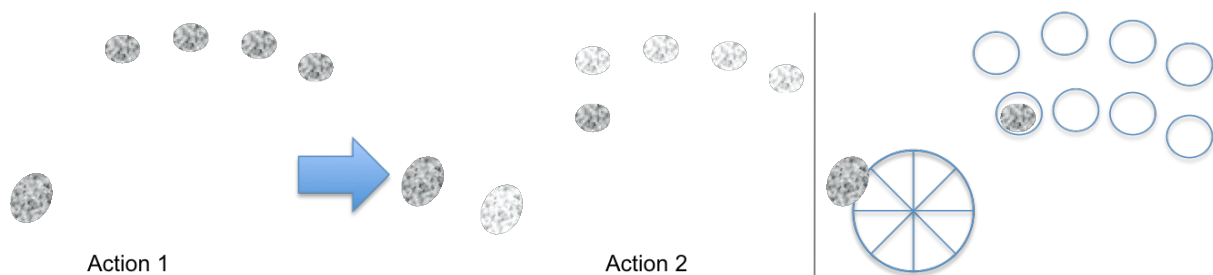


Figure 40 : Activation d'une commande en mode expert avec les cinq doigts comme type d'activateur. A gauche, l'utilisateur active le menu en posant les cinq doigts et fixe le référentiel. Au milieu, l'utilisateur pose le pouce et l'index de manière à activer la commande désirée. A droite, son interprétation.

2.5.2 Apprentissage et mémorisation

Technique

Le MTM est compatible avec l'interaction par *pointer et cliquer* ce qui facilite la prise en main du menu. Un test informel du MTM avec des utilisateurs qui n'avaient reçu aucune consigne a montré qu'ils étaient capables d'utiliser la technique mais avec un seul doigt : l'index. Cette attitude peut être mise en relation avec celle adoptée par les pianistes débutants qui

n'utilisent qu'un seul doigt. Une solution consisterait à jouer une vidéo [PatentApple 08] ou à dessiner une main fantôme temporairement au-dessus du menu afin de montrer l'utilisation optimale de la technique et de suggérer l'utilisation de plusieurs doigts.

Organisation

Le MTM est organisé hiérarchiquement. Ceci favorise les associations entre les éléments. Au second niveau de la hiérarchie, il est possible de faire des groupements internes comme le montre la Figure 41. De plus, le MTM possède les trois propriétés de navigation (la prévisualisation, la visibilité du chemin et l'accès direct aux parents) qui favorisent la formation d'une représentation mentale de la hiérarchie et l'apprentissage de la structure.

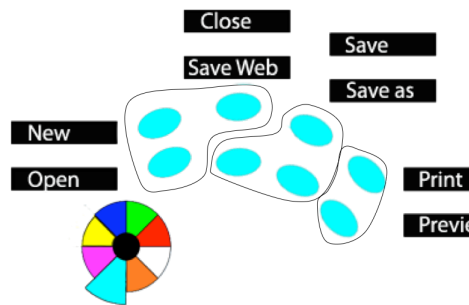


Figure 41 : Groupements internes dans le MTM.

Il convient de noter cependant que des groupements de plus de 2 items impliquent plusieurs doigts. Pour renforcer le lien entre les items, nous pourrions envisager que tous les items d'un même groupe soient associés au même doigt.

Mode expert

Un aspect important du Multi-Touch Menu est qu'il favorise une transition fluide du mode novice vers le mode expert. En effet, comme pour les Marking menus, les gestes sont identiques dans les deux modes de fonctionnement.

2.5.3 Satisfaction

Un test informel a permis de constater que la technique est bien acceptée par les utilisateurs malgré le fait qu'une fatigue du poignet apparaissait parfois car l'activateur utilisé reposait seulement sur l'utilisation du talon de la main dans l'expérience effectuée.



Nous retenons pour le facteur d'utilisabilité que le MTM permet la prévisualisation, l'accès direct aux commandes en mode expert en autorisant des gestes simultanés et la fluide transition fluide du mode novice vers le mode expert.

Nous étudions maintenant les propriétés du Multi-Touch Menu pour le facteur d'applicabilité.

2.5.4 Adéquation à l'application

La configuration par défaut du Multi-Touch Menu contient $8 \times 8 = 64$ commandes. Nous proposons plusieurs solutions pour augmenter le nombre de commandes.

En ne se limitant pas aux gestes droits, la première solution consiste à augmenter le premier niveau de la hiérarchie en acceptant les gestes courbes du Flower menu (chapitre 7). Ces gestes doivent rester suffisamment simples (droits ou inclinés) afin que les utilisateurs puissent les dessiner précisément avec le pouce.

Au second niveau de la hiérarchie, c'est-à-dire sous les doigts longs, nous proposons d'exploiter les mouvements d'abduction de l'index et de l'auriculaire (section 2.1). La Figure 42 montre que ces deux doigts peuvent accéder à 4 items différents. Ainsi, 12 items sont disponibles au second niveau de la hiérarchie.

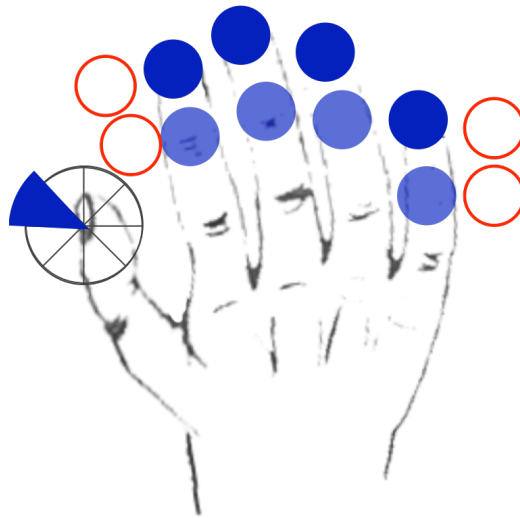


Figure 42 : Utilisation des mouvements d'abduction de l'index et de l'auriculaire pour accéder à 4 nouveaux items.

Enfin, chaque item sous les doigts longs peut donner accès à un nouveau menu circulaire (troisième niveau de la hiérarchie). Le Multi-Touch Menu pouvant ainsi contenir plusieurs centaines de commandes.

2.5.5 Adéquation à la plateforme

Le Multi-Touch Menu a été conçu pour des surfaces interactives multi-points de taille supérieur ou égale à celle de la main. Comme tous les périphériques directs, la main de l'utilisateur peut occulter des zones du menu et dégrader la recherche visuelle (section 2.5.1). Pour cette raison, nous proposons de séparer l'entrée (les boutons) de la sortie (les libellés) comme le montre la Figure 43. Même si les zones actives ne sont pas toujours visibles, l'utilisateur peut visualiser les libellés correspondants. Ces derniers sont affichés sous forme de bulle attachée à la zone active afin de mettre en évidence la relation entre ces éléments.

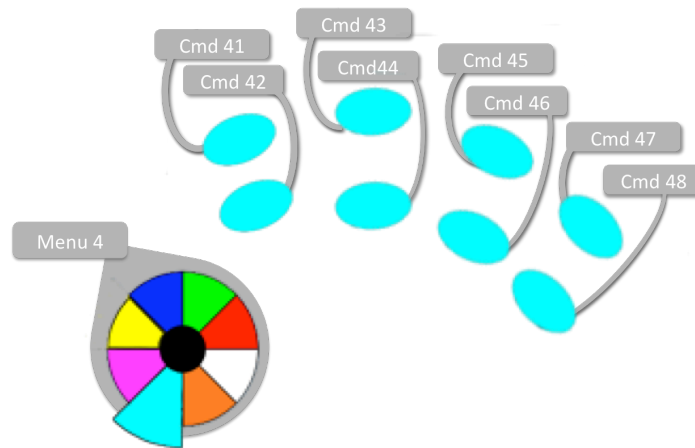


Figure 43 : Décalage des libellés des commandes pour éviter les problèmes d'occultation de la main sur le menu.

2.5.6 Adéquation à la tâche

Transitions

Transition Objet-Commande. L'activation par le talon n'est pas suffisamment précise pour activer le menu sur un petit élément de l'interface. Cependant, comme les surfaces multi-touch permettent l'interaction bi-manuelle, l'utilisateur pourrait pointer l'objet d'intérêt avec la main dominante et manipuler le menu contextuellement avec la main non-dominante.

L'activation avec les cinq doigts permet d'activer le menu contextuellement à un élément de l'interface. Lorsque l'utilisateur pose les cinq doigts, la position de l'index, doigt généralement utilisé pour pointer, définit l'objet servant de contexte au menu.

Transition Commande-Objet. La main de l'utilisateur ne se déplace pas pendant l'interaction avec le menu. Ainsi, à la fin de la sélection, la main est toujours proche du point d'activation.

Transition Commande-Commande. L'activation par le talon est particulièrement adaptée à la sélection multiple. Le MTM, comme la HotBox [Kurtenbach 99], reste ouvert tant que le talon de la main est en contact avec la table.

Le MTM permet également la *combinaison de commandes* grâce à l'interaction multi-points : l'utilisateur active plusieurs commandes simultanément. Habituellement, dans une application de dessin, lorsque l'on applique successivement deux couleurs à un objet, c'est la seconde couleur qui est affectée (la première étant remplacée). Avec le MTM, l'utilisateur pourrait « combiner » ou plutôt mélanger des couleurs. Par exemple, s'il active simultanément les commandes associées aux couleurs rouge et bleu (Figure 44), la couleur résultante sera violette (combinaison du rouge et du bleu).

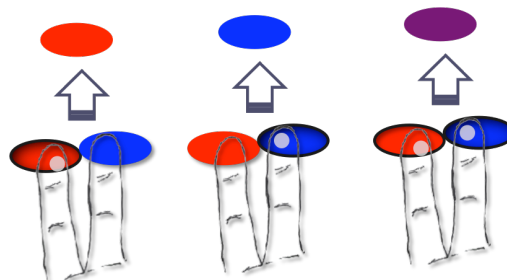


Figure 44 : Combinaison de commandes avec le MTM. Les deux commandes sont combinées pour mélanger les couleurs.

Transition Commande-Valeur. Le MTM permet le « merging », c'est-à-dire la sélection d'une commande puis le contrôle de sa valeur dans le même geste. Par exemple, lorsque l'utilisateur presse le bouton associé à la commande zoom, il peut déplacer son doigt dans l'axe de la main pour contrôler le niveau de zoom.

Contexte visuel

Le MTM permet la sélection sans regarder en exploitant la capacité des utilisateurs à discriminer leurs doigts. Les utilisateurs peuvent donc utiliser le MTM alors que leur attention est focalisée sur d'autres éléments de l'interface.



Nous retenons pour le facteur d'applicabilité que le MTM exploite le caractère multi-point de la surface pour combiner des commandes et permettre la sélection sans regarder grâce à la capacité des utilisateurs à discriminer leurs doigts.

2.6 Synthèse

Dans cette section, nous avons exploité les possibilités des surfaces interactives multi-points pour étudier la sélection de commandes. Nous avons proposé le Multi-Touch Menu, une technique d'interaction qui exploite les 5 cinq doigts de la main. Il dispose de deux types d'activateur originaux compatibles avec les techniques d'interaction existantes. Le premier utilise le talon de main, une partie rarement exploitée. Il est particulièrement adapté pour la répétition de commandes grâce à la propriété de persistance mais peut se révéler fatiguant pour le poignet à la longue. Le second utilise le contact simultané des cinq doigts. Il permet de positionner précisément les éléments du menu et est particulièrement adapté aux positions debout autour de la table. En exploitant plusieurs doigts, le Multi-Touch Menu permet la prévisualisation et la combinaison de commandes. Il permet également de traverser la hiérarchie de commandes rapidement en effectuant des gestes simultanés.

Nous envisageons de poursuivre les évaluations du Multi-Touch Menu à court terme. Dans un premier temps, nous étudierons les empreintes laissées par les utilisateurs lorsqu'ils utilisent les deux types d'activation afin d'optimiser le système de reconnaissance. Dans un second temps, nous comparerons le MTM avec d'autres techniques de menus.

3 Conclusion

Dans ce chapitre, nous nous sommes intéressés aux techniques de menus sur surfaces interactives considérant d'abord le cas des petites surfaces sur dispositifs mobiles puis celui des grandes surfaces comme celles des tables augmentées. Dans le cas des petites surfaces, après avoir étudié les contraintes des dispositifs mobiles et leurs impacts sur la conception de techniques de menus, nous avons proposé le Leaf menu. Dans le cas des grandes surfaces, nous avons tiré partie des nouvelles possibilités des surfaces multi-points afin de concevoir des menus efficaces en proposant le Multi-Touch Menu.

Le Leaf menu est un menu linéaire enrichi de raccourcis gestuels, ce qui le rend particulièrement bien adapté aux terminaux qui ne disposent pas de clavier comme les smartphones. Il permet la sélection d'items près des bords de l'écran en modes novice et expert. Son originalité réside dans l'introduction d'un effet miroir favorisant la sélection des items fréquents. Une évaluation préliminaire du Leaf menu semble indiquer que les utilisateurs ne sont pas gênés par cet effet et sont capables de mémoriser les différents gestes associés à une commande après peu de répétitions. Elle suggère encore une fois que la répétition n'est pas le seul facteur favorisant l'apprentissage du mode expert. La simplicité des gestes et leur représentation dans le menu sont également des facteurs essentiels pour le critère *apprentissage et mémorisation*.

La seconde technique de menus présentée est le Multi-Touch Menu qui repose sur les avantages de la technologie multi-points. Le MTM exploite les cinq doigts afin d'améliorer la sélection et la navigation dans une hiérarchie de commandes. Nous avons également proposé deux types d'activateurs compatibles avec les techniques d'interaction usuelles.

L'étude de ces deux menus nécessite d'être approfondie en réalisant une évaluation sur la capacité des utilisateurs à dessiner leurs marques respectives et en les comparant à d'autres techniques existantes. Ces études pourraient alors permettre d'enrichir MenUA de résultats spécifiques aux surfaces tactiles.

Conclusion

Chapitre 9

1	CONTRIBUTIONS DE LA THESE	227
1.1	CARACTERISATION DES MENUS : ESPACE MENUA	227
1.2	NOUVELLES TECHNIQUES DE MENUS	227
1.2.1	WAVE MENUS	227
1.2.2	FLOWER MENUS	228
1.2.3	LEAF MENUS	228
1.2.4	MULTI-TOUCH MENUS	229
2	PERSPECTIVES A MOYEN TERME	229
2.1	MENUA	229
2.2	PLATEFORME POUR LA REALISATION DE MENUS	230
3	PERSPECTIVES A LONG TERME	233

En conclusion, nous résumons nos contributions qui concernent la caractérisation, la conception et l'évaluation des techniques de menus. Une prise de recul par une analyse des résultats nous permet ensuite d'identifier plusieurs perspectives de développement à moyen et long terme.

1 Contributions de la thèse

Les deux facettes complémentaires de nos contributions sont la caractérisation des techniques de menus grâce à notre espace MenUA et la définition de quatre nouvelles techniques de menus.

1.1 Caractérisation des menus : espace MenUA

Face à la difficulté de caractériser les techniques de menus et de comprendre les propriétés interactionnelles impliquées, nous avons proposé un espace de conception des techniques de menus, appelé **MenUA**. Nous avons établi une liste de critères utiles pour étudier finement les avantages et les inconvénients respectifs des techniques. Pour chaque critère, nous avons explicité, décrit et comparé des solutions que nous avons illustrées par les techniques de la littérature. La contribution de l'espace MenUA revêt plusieurs facettes :

- Dresser un état de l'art des travaux sur les menus. Notre synthèse offre une vue globale et structurée du domaine grâce aux critères que nous avons proposés. MenUA par exemple a été utilisé pour l'enseignement de l'IHM à Toulouse afin « d'illustrer les différentes techniques d'optimisation de la sélection d'items dans un menu ».
- Favoriser l'introduction des nouveaux menus dans les applications. MenUA présente aux concepteurs d'applications les techniques existantes et leur permet de faire des choix informés pour introduire le menu le plus adapté à l'application considérée.
- Etre un outil pour la conception de nouveaux menus. En mettant en évidence des éléments moins étudiés de l'espace, MenUA peut aider les concepteurs à améliorer et créer de nouvelles techniques. Il a contribué à l'élaboration de quatre nouvelles techniques de menus présentées dans la seconde partie du manuscrit. De plus, MenUA permet de guider la conception en s'appuyant sur des critères et leurs aspects associés.
- Favoriser l'évaluation d'une technique. MenUA a facilité la comparaison analytique de nos techniques aux techniques existantes en les caractérisant par des propriétés interactionnelles. Enfin, il a aidé la mise en place de nos évaluations expérimentales en identifiant précisément les critères que nous souhaitons étudier.

1.2 Nouvelles techniques de menus

Nous avons conçu, développé et évalué expérimentalement quatre techniques de menus. Ces techniques traitent des aspects différents de notre espace MenUA.

1.2.1 Wave menus

Les utilisateurs doivent souvent naviguer dans un système de menus pour trouver la commande désirée. Cependant, les chemins détournés n'ont pas reçu une grande attention dans les travaux sur les menus. En particulier, les études sur les Marking menus ont surtout mis l'accent sur le mode expert et non la navigation/exploration en mode novice.

Nous avons d'abord souligné l'importance de l'efficacité d'une technique pour les chemins détournés. Nous avons ensuite explicité les principales solutions interactionnelles qui aident la navigation en mode novice, en particulier la *prévisualisation*. Puis, nous avons présenté

une nouvelle technique de Marking menus appelée Wave menus. L'originalité de celle-ci repose sur une organisation concentrique inversée permettant d'exploiter au mieux l'espace disponible et qui favorise la navigation dans la hiérarchie de commandes.

Une évaluation avait comparé deux Marking menus, le Multi-Stroke menu (MSM) et le Compound Marking menu (CMM), en mode expert [Zhao 04]. Les résultats montraient que le MSM était plus efficace que le CMM en mode expert. Cependant, ces menus n'avaient pas été comparés en mode novice. Pour cette raison, nous avons proposé une expérimentation contrôlée visant à évaluer ces menus hiérarchiques ainsi que le Wave menu en mode novice. Face à la difficulté de ce type d'expérimentation, nous avons proposé une tâche de navigation paramétrable que nous avons utilisée pour simuler des chemins détournés. Les résultats montrent que, contrairement à l'évaluation du mode expert, les CMM sont plus efficaces que les MSM en mode novice. De plus, les Wave menus sont plus efficaces que les MSM pour notre tâche de navigation.

1.2.2 Flower menus

Les gestes des marking menus ne permettent pas de sélectionner précisément un grand nombre de commandes en largeur. Face à ce constat, nous avons proposé un nouveau vocabulaire gestuel, exploitant en plus de l'orientation, la courbure et le sens, afin de vérifier la capacité des utilisateurs à exécuter ces gestes correctement. Ces gestes courbés sont exploités dans une nouvelle technique de menus, notée Flower menus. Les Flower menus améliorent les Marking menus :

- en augmentant le nombre de commandes maximum en largeur ;
- en mettant en valeur les groupements internes sous la forme visuelle d'une branche.

Dans le contexte de nos travaux sur les Flower menu, nous avons souligné l'importance du critère *apprentissage et mémorisation*, qui a été relativement peu étudié jusqu'à présent, et montré que la répétition n'est pas le seul facteur favorisant ce critère. Nous avons défini une tâche pour l'*apprentissage explicite* du mode expert que nous avons utilisée pour comparer expérimentalement les Flower menus, les Polygon menus et les menus linéaires. Les résultats montrent que les Flower menus favorisent l'apprentissage du mode expert.

1.2.3 Leaf menus

Face aux contraintes de l'interaction sur dispositifs mobiles (petit écran, absence de clavier, imprécision, etc.), nous avons proposé une nouvelle technique de menus, notée Leaf menus, conçue spécifiquement pour ce type de plateformes. Le Leaf menu est un menu linéaire enrichi de raccourcis gestuels basés sur le vocabulaire du Flower menu. Une originalité du Leaf menu est d'utiliser un effet miroir qui inverse l'ordre d'affichage des items lorsque le menu est activé en bas de l'écran. Cet effet miroir a l'avantage de favoriser la sélection des items favoris en mode novice et de permettre leur sélection près des bords de l'écran en mode expert grâce à la redondance gestuelle.

Une évaluation préliminaire a montré que les utilisateurs n'étaient pas gênés par l'effet miroir et étaient capables de mémoriser les raccourcis gestuels après très peu de répétitions. Une évaluation plus approfondie du Leaf menu constitue une perspective à court terme. L'expérimentation viserait à comparer les performances du Leaf menu avec un Leaf menu sans redondance gestuelle et un Marking menu classique pour une tâche d'apprentissage et de mémorisation. L'objectif serait d'étudier l'impact de la redondance gestuelle et du couple (geste/position de l'item) sur le critère apprentissage et mémorisation.

1.2.4 Multi-Touch Menus

Les techniques de menus exploitent peu les possibilités de la main. Ce constat est sans nul doute dû aux limitations des dispositifs d'entrée. Aujourd'hui, les surfaces multi-points offrent de nouvelles possibilités. Cependant, la plupart des nouvelles techniques proposées pour ces surfaces reposent uniquement sur l'utilisation de deux ou trois doigts.

Basé sur un modèle de la main comprenant 21 degrés de liberté, nous avons proposé une nouvelle technique de menus exploitant les cinq doigts de la main : le Multi-Touch Menu. Deux types d'activateur ont été définis : l'un basé sur le talon de la main, l'autre sur la pose simultanée des cinq doigts.

L'évaluation du MultiTouch Menu constitue la seconde perspective à court terme. L'expérimentation sera décomposée en deux étapes. La première aura pour objectif d'obtenir des données expérimentales pour améliorer notre système de reconnaissance du talon de la main et de la pose simultanée des cinq doigts. La seconde étape consistera à comparer le Multi-Touch Menu à un Marking menu. Nous envisageons une tâche permettant d'étudier leurs performances respectives pour les trois types de chemins (détourné, droit et direct). Nous utiliserons un petit nombre de commandes à sélectionner (~10) comme stimuli. L'utilisateur devra d'abord naviguer pour les trouver, puis avec l'expérience, il pourra y accéder sans détour et enfin y accéder en mode expert. A notre connaissance, une expérimentation avec 3 types de chemins n'a jamais été conduite. Elle permettra d'analyser les performances pour chacun des chemins ainsi que l'évolution au cours du temps.

Les perspectives de travaux futurs sont nombreuses et divisées en deux parties : les perspectives à moyen terme et long terme.

2 Perspectives à moyen terme

Nous retrouvons dans nos perspectives de développement, les deux facettes de nos travaux que sont la caractérisation des techniques d'interaction et la définition de nouvelles techniques de menus.

2.1 MenUA

Deux voies ont été envisagées pour enrichir MenUA : son extension à l'évaluation prédictive et sa généralisation aux techniques d'interaction.

Extension : évaluation prédictive

MenUA permet de comparer qualitativement les menus. Une perspective consiste à compléter notre espace par des modèles de performance. Nous envisageons d'étendre le modèle de Cockburn [Cockburn 07] sur la performance des menus linéaires en mode novice en considérant :

- le mode expert des menus linéaires ;
- le contexte dans lequel ils sont utilisés [Appert 07] ;
- les menus circulaires ;
- l'influence du vocabulaire gestuel utilisé [Cao 07] ;
- l'impact de solutions présentées dans MenUA comme la prévisualisation.

Généralisation aux techniques d'interaction

Une autre perspective de travail a trait à la *généralisation* de MenUA. Initialement centré sur les techniques de menus, nous envisageons de l'étendre aux techniques d'interaction. En effet, les critères utilisés ainsi que la plupart des *solutions* présentées ne sont pas spécifiques aux menus et peuvent être généralisés aux techniques d'interaction. Les *aspects*, traits communs à

toutes les techniques sur le quel on peut agir vis-à-vis d'un critère, devront être redéfinis afin de prendre en compte les spécificités du groupe de techniques considéré.

Par exemple, la sélection de petites cibles sur dispositifs mobiles est un problème délicat [Roudaut 08], auquel nous avons été sensibilisé durant la thèse. Comme pour les techniques de menus, il existe de nombreuses techniques de pointage dont il est difficile de comprendre les similitudes et les différences. Nous envisageons d'étudier ces techniques de pointage vis-à-vis de MenUA en commençant par identifier les différents aspects à traiter puis en étudiant dans quelle mesure les solutions déjà proposées peuvent s'appliquer.

2.2 Plateforme pour la réalisation de menus

Nos travaux nous ont amenés à concevoir et développer de nombreuses techniques de menus : les nouvelles techniques proposées que sont le *Wave Menu* (chapitre 6), le *Flower menu* (chapitre 7), le *Leaf Menu* et le *Multi-Touch Menu* (chapitre 8), mais aussi les techniques utilisées lors des évaluations expérimentales que sont le *Marking menu*, le *Multi-Stroke menu*, le *Polygon menu* et le *menu linéaire*, ainsi que d'autres prototypes. Nous visons une capitalisation de cette expérience au sein d'une plateforme logicielle pour la conception, le développement et l'évaluation des techniques de menus. Nous justifions d'abord son intérêt scientifique puis nous présentons des pistes pour sa réalisation.

Pour la conception

Il est regrettable de ne pas disposer d'une bibliothèque de menus, et plus largement de techniques d'interaction [imuseum], afin de pouvoir essayer ces techniques et apprécier finement leurs détails interactionnels. Ainsi, les concepteurs de menus n'ont pas tous les éléments pour comprendre précisément le fonctionnement d'un menu et identifier les inconvénients ou les cas limites d'utilisation. Un objectif de cette plateforme consisterait à pouvoir tester des menus afin de faciliter le travail des concepteurs. Evidemment, la plateforme n'a de sens que si un grand nombre de techniques sont disponibles. Aussi, il est nécessaire de faciliter leur développement.

Pour le développement

Nous avons constaté durant nos travaux que le développement de menus hiérarchiques est complexe. Par exemple, la classe « QMenu » correspondant au menu linéaire de la boîte à outils *Qt*, comprend plus de 3000 lignes de code. Or, le développement d'un *Marking menu* nécessite davantage de programmation de par sa représentation circulaire, sa gestion des évènements, et son système de reconnaissance gestuelle.

Les boîtes à outils actuelles (*Qt*, *Swing*, etc.) ne sont pas adaptées aux développements de ces techniques d'interaction [Appert 06c, Appert 07] car les composants graphiques sont difficilement réutilisables). Les boîtes à outils "nouvelle génération" [Lecolinet 03, Dragicevic 04, Blanch 05, Appert 07], conçues pour dépasser les précédentes limitations, sont généralement peu utilisées. Les développeurs de menus sont alors contraints de fournir un effort de programmation important qui n'est pas toujours compatible avec des travaux de recherche dans un temps limité (thèse, post-doc). Il est alors fréquent que les techniques ne soient développées que partiellement (par exemple uniquement le mode expert) afin de limiter le temps de développement. Le risque est de ne pas identifier d'éventuels problèmes de conception comme des incohérences avec le mode novice. Un objectif de notre plateforme consisterait à faciliter le travail des développeurs en proposant un ensemble de mécanismes de base réutilisables et adaptés à ces techniques.

Pour l'évaluation

Comme le concepteur de nouvelles techniques ne dispose pas des techniques qu'il souhaite comparer, il est contraint de les développer et il n'a pas la garantie que le résultat soit conforme à la technique originelle. Or, des détails interactionnels, comme les délais des menus linéaires, peuvent être déterminants pour les performances. Un objectif de notre plateforme consisterait à faciliter la réplication d'expériences et la comparaison de techniques grâce à un « benchmark » de techniques servant de base de comparaison et d'un système de capture de traces (« logs ») uniforme.

Vers la réalisation de la plateforme

Les techniques développées pendant la thèse nous ont montré l'utilité d'avoir un outil fonctionnel d'*introspection* pendant la phase de conception. Cet outil facilite le prototypage rapide en permettant de manipuler aisément les propriétés du menu (positionnement des items, couleurs, délai, etc.).

Une attention particulière devra être portée à la gestion des événements. Ces derniers sont délicats à traiter car les menus sont « au-dessus » de l'application et se comportent différemment des autres composants graphiques. Notre solution a consisté à utiliser des machines à état lesquelles facilitent grandement le développement de ces techniques. Cependant, la réutilisation d'une machine à état d'une technique à une autre reste un problème délicat. Pour cela, nous envisageons d'approfondir les machines à états hiérarchiques [Blanch 05] et de nous tourner vers des outils tels que *SwingStates* [Appert 07]. Cet aspect est essentiel car une partie importante du développement d'un menu consiste à coder la partie « contrôle ». Cette partie constitue, à notre avis, le problème le plus difficile et le plus intéressant à résoudre.

Enfin, nous envisageons également d'enrichir la plateforme d'outils permettant de faciliter la mise en place d'évaluations contrôlées et l'analyse statistique des résultats [Mackay 07, Guimbretière 07].

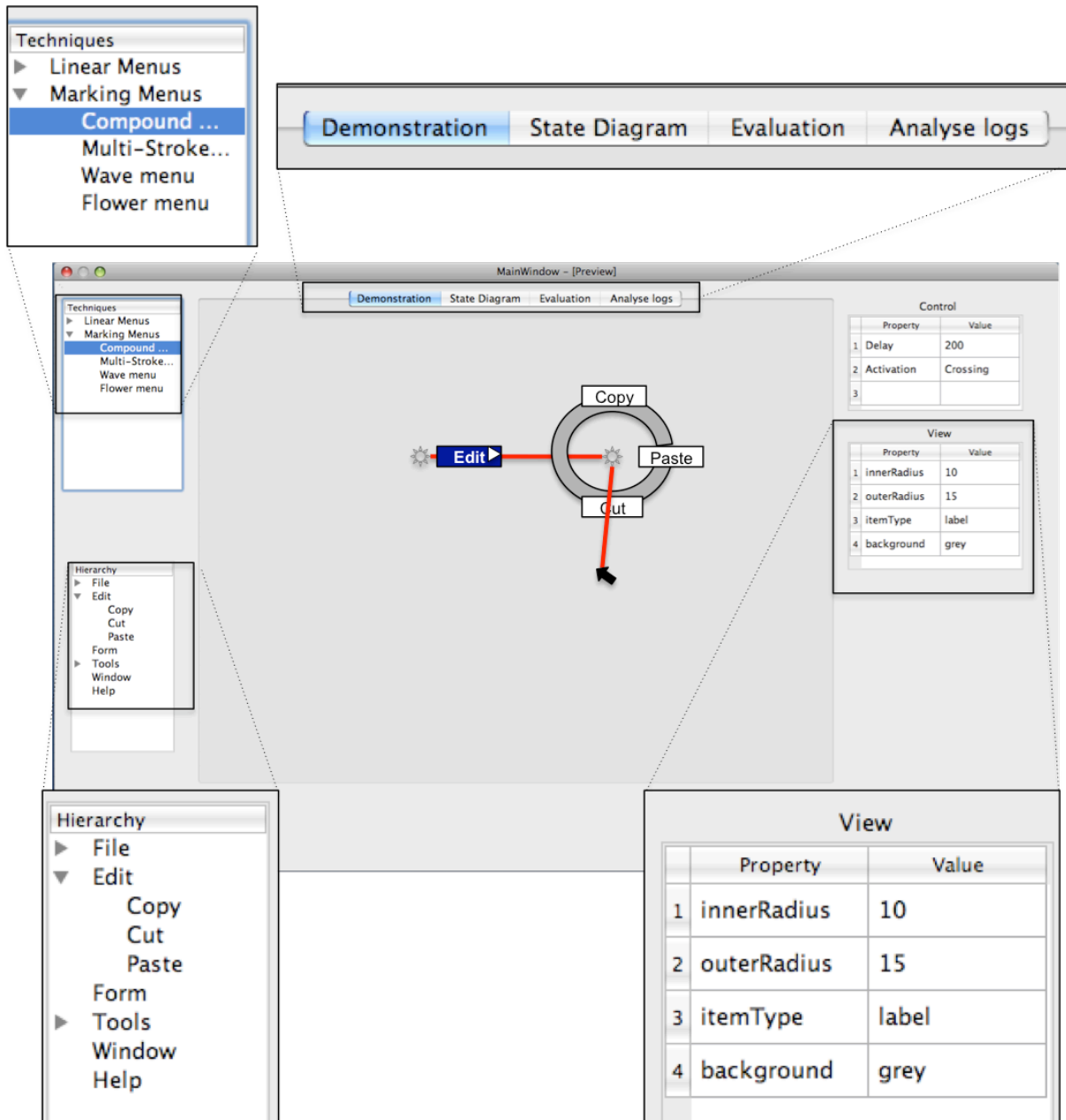


Figure 1 : Maquette de la plateforme de menus. Au centre, une zone pour essayer une technique de menus. En haut à gauche, les différentes techniques de menus disponibles. En haut à droite, différents onglets permettant de tester la technique (« Demonstration »), de manipuler l'automate (« state diagram »), d'évaluer des techniques (« évaluations ») ou d'analyser les traces d'exécution (« analyse logs »). En bas à gauche, la hiérarchie de commandes que l'utilisateur peut éditer. Enfin, en bas à droite, des propriétés graphiques que l'utilisateur peut manipuler dynamiquement.

3 Perspectives à long terme

Les chapitres 4 et 7 ont abordé le critère *apprentissage et mémorisation*. Celui-ci a moins été étudié dans la littérature que celui de *rapidité et précision* vis-à-vis des techniques d'interaction en particulier celles basées sur les gestes. Cependant, nous pensons que ce critère est essentiel pour l'acceptabilité d'une technique d'interaction. En effet, l'utilisateur doit pouvoir rapidement acquérir le savoir nécessaire pour utiliser le menu de manière optimale.

Aujourd'hui, les performances en vitesse des techniques de menus sont bonnes et les améliorations permettent de gagner seulement des dixièmes de seconde. Au contraire, le gain en temps nous semble beaucoup plus important en facilitant l'apprentissage de l'organisation des menus et l'acquisition du mode expert.

Pour traiter ce critère *apprentissage et mémorisation* qui nous semble aujourd'hui plus important que celui de *rapidité et précision*, des travaux pluridisciplinaires doivent être entrepris, en particulier avec les sciences cognitives. En effet affiner le critère *apprentissage et mémorisation* en sous-critères nécessite de s'appuyer sur des modèles de sciences cognitives. Ces modèles permettraient d'identifier les propriétés interactionnelles impliquées et de guider la conception par une analyse systématique des différents aspects de ce critère.

Pour démarrer ce travail, nous envisageons de nous concentrer sur le couple « geste et mémorisation » en étudiant l'impact du vocabulaire de gestes, des associations (gestes/commandes) et de la répétition sur la mémorisation d'un ensemble de commandes. Il existe un grand nombre de questions ouvertes qu'il serait intéressant d'investiguer, en particulier :

- Quels sont les critères pour affecter un geste à une commande ?
- Quel impact a le fait d'associer des gestes « proches » (dans l'exécution du geste) à des commandes proches sémantiquement ? Cet aspect favorise-t-il l'apprentissage par l'utilisation d'associations ? Comment gérer alors la cohérence globale pour un plus grand nombre de commandes ?
- Comment représenter les gestes dans un menu ? [Appert 09]

Voici, un ensemble de questions pour lesquelles il serait intéressant d'explorer différents types d'association (commande /geste/ représentation du geste) afin d'identifier les facteurs ayant un impact important sur les techniques de menus.

Tables des figures

Chapitre 2

Figure 1 : Illustration extraite de [Shneiderman 92]	11
Figure 2 : Démarche d'analyse et structure du chapitre.....	12
Figure 3 : Modèle de l'interaction instrumentale (illustration extraite de [Beaudouin-Lafon 00]).	13
Figure 4 : Menu dont les éléments sont discrets (commandes).....	15
Figure 5 : Menu dont les éléments sont continus (le niveau sonore) dans un ensemble borné..	15
Figure 6 : Ligne de commande qui force les utilisateurs à se rappeler le nom et la syntaxe de la commande.....	16
Figure 7 : Relations entre le menu courant, un super menu (ou menu parent) et un sous-menu.	17
Figure 8 : A gauche les Tracking menus [Fitzmaurice 03], à droite les Pie Cursors [Fitzmaurice 08] (illustrations extraites de [Fitzmaurice 08]).	18
Figure 9 Combobox.....	18
Figure 10 : 4 types de menus. (a) Les « Pull-down menus », (b) Les « Cascading menus », (c) « Pop-up menus », (d) « Tear-off menus ».	19
Figure 11 : A gauche un menu linéaire et à droite un menu circulaire (illustration extraite de [Callahan 88]).	20
Figure 12 : Mnémonique (la lettre « p ») et raccourci clavier (Ctrl+X) pour la commande « Couper » du menu « Edition ». La mnémonique peut être utilisé uniquement lorsque le menu est ouvert en appuyant sur la lettre correspondante. Au contraire, le raccourci clavier peut être utilisé même lorsque le menu est fermé. Il est unique dans tout le système de menus.....	21
Figure 13 : Marking menus classiques [Kurtenbach 91]. A gauche, le mode novice. Après 300ms, le menu s'affiche et l'utilisateur sélectionne une commande comme dans les menus linéaires. Si l'utilisateur n'attend pas ce délai, le menu est en mode expert (à droite). Le menu n'est pas affiché et le curseur laisse une trace dont l'orientation est interprétée par le système (Illustration extraite de [Kurtenbach 91]).	22
Figure 14 : Interaction avec un menu : couplage des deux points de vue, système et utilisateur..	26
Figure 15 : Différents types de chemins pour accéder à une commande. A) chemin fortement détourné, b) chemin faiblement détourné, c) chemin droit, d) chemin direct.....	26
Figure 16 : Couverture et mode de fonctionnement associés à un chemin fortement détourné. .	27
Figure 17 : Couverture et mode de fonctionnement associés à un chemin faiblement détourné.	28
Figure 18 : Couverture et mode de fonctionnement associés au chemin droit.	29
Figure 19 : Couverture et mode de fonctionnement associés au chemin direct.	29

Chapitre 3

Figure 1 : Arbre de l'acceptabilité d'un système (illustration extraite de [Nielsen 93]).	34
Figure 2. Correspondance entre les cinq critères d'utilisabilité de Shneiderman [Shneiderman 04] et Nielsen [Nielsen 93]. Illustration extraite de [Welie 99b].	35
Figure 3 : Graphe d'interaction pour un menu contextuel (haut) et une Toolglass (bas) qui ont trois commandes (créer triangle, créer rectangle, créer ellipse). Illustration extraite de [Appert 04].	36
Figure 4 : Largeur et distance d'un item pour la loi de Fitts.	38
Figure 5 : 18 « menus 3D » pour les environnements de RA et VR. Illustrations extraites de [Daschselt 07].	40
Figure 6 : Classification des solutions pour les « menus 3D » en fonction de critères (en gras) et de propriétés (en italique). Illustration extraite de [Dachselt 07].	41

Chapitre 4

Figure 1 : Processus pour l'exécution d'une commande dans un menu.	53
Figure 2 : Les Frequency-ordered menus [Lee 04]. Les items 4 et 3 sont les plus utilisés et sont donc déplacés en haut du menu. Illustration extraite de [Lee 04].	54
Figure 3 : Folded menus. Initialement replié pour laisser apparaître uniquement les items les plus fréquents, le menu est déplié après un délai ou un clic sur le bouton « dépliage ». (Illustration extraite de [humanf. 04])	55
Figure 4. Mise en valeur des items fréquents en exploitant des attributs visuels. A) les Morphing menus [Cockburn 07] qui augmentent la hauteur des items, b) les Bubbling menus [Tsandilas 07] qui changent la couleur du fond, c) menus linéaires avec des icônes et d) les menus translucides [Bowes 03] qui augmentent la transparence des items les moins fréquents.	55
Figure 5 : Les Temporal menus [Lee 04] affichent d'abord les items fréquents sans modifier l'organisation des commandes. Illustration extraite de [humanf. 04].	56
Figure 6 : Les Split menus [Sears 94]. Les items les plus fréquents sont dupliqués et positionnés en haut du menu. Illustration extraite de [Sears 94].	56
Figure 7 : Groupements internes dans les menus linéaires séparés par une ligne.	57
Figure 8 : Pie menu [Callahan 88]. Les items sont organisés circulairement autour du point d'activation. Illustration extraite de [second Life].	59
Figure 9 : Réduction en moyenne de la distance des items en centrant les menus linéaires à mi-hauteur.	59
Figure 10 : L'utilisateur réalise un mouvement sur la droite pour ouvrir le sous-menu. Celui-ci s'affiche sous le curseur. Illustration extraite de [Koybayashi 03].	60
Figure 11 : La hauteur des items augmente proportionnellement à la distance afin de conserver un indice de difficulté constant.	60
Figure 12 : Indépendance à l'échelle dans les Marking menus [Kurtenbach 91]. La taille des marques n'a pas d'importance.	61

Figure 13 : Le chemin le plus court pour accéder à un item du sous-menu est la diagonale. Sans l'utilisation, d'un délai, ce chemin provoquerait la fermeture du sous-menu. Illustration extraite de [Cockburn 96].	61
Figure 14 : MenUs [Cockburn 06] laissent le sous-menu affiché tant que le curseur n'est pas dans un nouvel item hiérarchique. La zone d'ouverture du menu est donc élargie . Illustration extraite de [Cockburn 06].	62
Figure 15 : Les AAMU menus [Tanvir 08]. Introduction d'une zone triangulaire pour faciliter l'accès en diagonale aux sous-menus. De plus les sous-menus sont affichés à mi-hauteur de l'item parent. Illustration extraite de [Tanvir 08].	62
Figure 16 : Pointage sémantique. Le ratio « control/display » comme facteur d'échelle entre l'écran (ligne de pixels en haut) et l'espace de la souris (« images » des pixels sur la table après un changement d'échelle en bas). Illustration extraite de [Blanch 04a].	63
Figure 17 : (a) Version à l'écran, (b) Version dans l'espace moteur qui permet de réduire les entrées inactives. Illustration extraite de [Blanch 04b].	63
Figure 18 : Les Force Fields menus [Ahlström 05] utilisent un « champs de force » pour favoriser l'accès aux sous-menus et prévenir des erreurs. Illustration extraite de [Ahlström 05].	63
Figure 19 : Les Jumping menus [Ahlström 06] qui permettent de faire sauter le curseur d'un item parent à son sous-menu. Illustration extraite de [Ahlström 06].	64
Figure 20 : Le Bubble curseur. La taille du curseur change dynamiquement afin de toujours sélectionner la cible la plus proche ($d_1 < d_2$). Illustration extraite de [Grossman 05].	64
Figure 21 : Les Bubbling menus [Tsandillas 07]. (a) La vue par défaut avec les items favoris en bleu (bouton de la souris relevé). (b) Utilisation du Bubble curseur qui saute de favoris en favoris (bouton de la souris enfoncé). (c) Lorsque le curseur revient dans la partie gauche du menu, le Bubble curseur disparaît. Illustration extraite de [Tsandillas 07].	65
Figure 22 : Les menus rythmiques [Maury 99]. Le système met en surbrillance chaque item à rythme constant. Illustration extraite de [Maury 99].	65
Figure 23 : Dans les Compound Marking menus, la marque en mode expert correspond au chemin à parcourir en mode novice.	66
Figure 24 : Les Multi-Stroke menus [Zhao 04]. En mode expert, les utilisateurs dessinent une série de marques simples correspondant au geste pour chaque sous-menu. Illustration extraite de [Zhao 04].	67
Figure 25 : Plusieurs solutions pour améliorer l'apprentissage des raccourcis claviers. A) conception classique, b) ajout d'un retour visuel après activation, c) remplacement du nom de la commande par le raccourci clavier lorsque le curseur est au dessus de l'item, d) utilisation d'un retour audio, e) introduction d'un délai, f) désactivation de la sélection par le menu. Illustration extraite de [Grossman 07].	73
Figure 26 : A gauche, le clavier Logickeyboard adapté pour le logiciel Final Cut [Logickeyboard]. A droite, le clavier Optimus [Optimus] où chaque touche est un petit écran.	74
Figure 27 : Construction visuelle d'un symbole chinois à partir de trois marques d'un Multi-Stroke menu. Illustration extraite de [Zhao 07b].	75
Figure 28 : Transition fluide du mode novice au mode expert pour les Marking menus [Kurtenbach 91] : les gestes sont identiques en mode novice (à gauche) et en mode expert (à droite). Illustration extraite de [Kurtenbach 91].	75
Figure 29 : Après l'activation d'une commande avec les Marking menus, la marque parfaite est dessinée quelques dixièmes de secondes. Illustration extraite de [Tapia 95].	75

Chapitre 5

- Figure 1 : L'amplitude des secteurs angulaires pour un menu circulaire avec 16 commandes.....85
- Figure 2 : Les Zone Menus [Zhao 06]. A gauche, le mode novice : l'utilisateur fait un clic (appui - relâchement) pour ouvrir le menu. L'écran est partagé en 4 zones qui donnent chacune accès à un Marking menu de 8 items (ici 4). L'utilisateur doit d'abord se déplacer dans la zone du Marking menu qui l'intéresse puis faire une marque partant du centre de ce menu et en direction de l'item désiré. A droite, le mode expert : L'utilisateur fait la même marque qu'en mode novice (un clic puis une marque relativement au point d'activation). Illustration extraite de [Zhao 06].85
- Figure 3 : Les Polygon menus (PM) [Zhao 06]. A gauche le mode novice. A droite, le mode expert. Ces menus fonctionnent comme les Zone menus : l'utilisateur doit faire un clic pour ouvrir le menu puis exécuter une marque en direction de l'item désiré relativement au point d'activation initial. Illustration extraite de [Zhao 06].86
- Figure 4 : Les DartBoard menus exploitent à la fois l'orientation et la distance. A gauche, illustration extraite de [Kurtenbach 93b] avec 4*4 items. A droite, illustration extraite de [Nancel 08] avec 19 items.86
- Figure 5 : Les Donuts menus [Lai 05] exploitent la pression pour augmenter le nombre de caractères chinois dans un Pie-menu. Illustration extraite de [Lai 05].86
- Figure 6 : Les Push Menus [Huot 08] exploitent la pression pour augmenter le nombre d'items en largeur dans un Marking menu. Illustration extraite de [Huot 08].87
- Figure 7 : Une vue en œil de poisson déformante [Furnas 86]. Le résultat (à droite) est obtenu à partir d'une vue initiale (à gauche) et un centre d'intérêt (en plus gros).....87
- Figure 8 : Le Fisheye menu [Bederson 00]. Le même menu affiché avec le curseur à trois positions différentes. Illustration extraite de [Bederson 00].88
- Figure 9 : FishTree menus [Lecolinet 06]. A) l'utilisateur zoome sur l'item « post_graduate ». b) la taille de cet élément grossit et son contenu commence à apparaître en transparence. c) à la fin de l'opération, l'intitulé initial est entièrement remplacé par le contenu du nœud. Illustration extraite de [Lecolinet 06].88
- Figure 10 : Un Compound Marking menu de profondeur 3 (présentée dans la section 1.5 du chapitre 4). A gauche le mode novice. A droite, le mode expert : une marque composée de plusieurs inflexions (composition spatiale). Illustration extraite de [Zhao 04]89
- Figure 11 : Ambigüités des marques avec les Compound Marking menus.89
- Figure 12 : Les Multi-Stroke menus [Zhao 04] (présentée dans la section 1.5 du chapitre 4). . A gauche, le mode novice. Les sous-menus se superposent. A droite, le mode expert. L'utilisateur fait une série de marques simples qui peuvent être superposées (composition temporelle). Illustration extraite de [Zhao 04]90
- Figure 13 : Menus circulaires avec des portions linéaires [Kurtenbach 97]. Illustration extraite de [Kurtenbach 97].90
- Figure 14 : La Hotbox [Kurtenbach 99]. Lorsque l'utilisateur appuie sur la barre d'espace, des barres de menus s'affichent au centre de l'écran et l'écran est partagé en 4 zones qui donnent accès à différents Marking menus. Illustration extraite de [TutorielMaya].91
- Figure 15 : Menus linéaires augmentés de raccourcis gestuels. Illustration extraite de [Kurtenbach 93b]93

Figure 16 : Occultation de la main sur le menu. A gauche, un menu linéaire, à droite, un Marking menu.....	93
Figure 17 : Imprécision du pouce pour sélectionner précisément un item dans un menu sur dispositif mobile. Illustration extraite de [Huot 07].....	94
Figure 18 : A gauche, ArchMenu ; a droite, ThumbMenu [Huot 07]. Ces deux menus utilisent le mécanisme de projection du curseur (« offset cursor ») pour éviter les problèmes d'occultation et d'imprécision (illustration extraite de [Huot 07]).	94
Figure 19 : Un Compound Marking menu à 3 niveaux occupe trois fois l'espace horizontal qu'un menu linéaire.....	95
Figure 20 : En absence de bord, les Marking menus sont affichés centrés autour du point d'activation. En présence de bord, le menu risque de déborder de l'écran. Une solution est le débrayage, mais il a l'inconvénient de rompre l'association commande-geste.....	96
Figure 21 : En affichant seulement les labels, les Marking menus masquent moins le contexte visuel de l'utilisateur. Illustration extraite de [Tapia 95].....	97
Figure 22 : Les items parents sont masqués dans certaines implémentations des Marking menus. Illustration extraite de [Tapia 95].	98
Figure 23 : Utilisation de la transparence dans le menu linéaire. La palice utilisée ici est appelée « l'anti-interférence ». Illustration extraite de [Harrison 96].....	98
Figure 24 : les différentes transitions impliquées dans le processus de sélection d'une commande entre le menu et le contexte.....	99
Figure 25 : Transition commande-objet avec les Compound Marking menus [Kurtenbach 93a]. Après avoir sélectionné Orange, l'utilisateur doit parcourir une distance importante pour revenir à l'objet d'intérêt (schématisé par le rectangle bisoté).....	100
Figure 26 : Transition commande-objet avec les Slippery menus [Cance 06]. Ce menu glisse sous le curseur afin que celui-ci reste à la position de départ (au dessus de l'objet d'intérêt)...	100
Figure 27 : Transition commande-objet avec les Multi-Stroke menus [Zhao 04]. Les sous-menus se superposent afin que le curseur ne s'éloigne jamais de l'objet d'intérêt.	100
Figure 28 : Les Floating Pie menus [Rubio 02]. L'utilisateur peut empêcher la fermeture d'un sous-menu grâce à un bouton de « lock » au centre du menu. Illustration extraite de [Rubio 02].....	101
Figure 29 : Dans CrossY [Apitz 04], les utilisateurs peuvent sélectionner plusieurs commandes en un seul geste grâce à la « composition fluide de commandes ». Illustration extraite de [Apitz 04].....	102
Figure 30 : Control menus [Pook 00]. L'utilisateur sélectionne la commande zoom en faisant un geste en direction de l'item. Un fois une distance seuil dépassée, la commande est activée et l'utilisateur contrôle continûment le niveau de zoom jusqu'à ce qu'il relâche le bouton de la souris. Illustration extraite de [Pook 00].....	102
Figure 31 : Flowmenus [Guimbretiere 00]. Après avoir sélectionné la commande « Move », l'utilisateur revient dans la zone centrale pour déplacer l'objet d'intérêt. Illustration extraite de [Guimbretiere 00].....	102
Figure 32 : Tilt menus [Tian 08]. L'utilisateur sélectionne une commande en inclinant le stylet dans une direction. Une fois la commande sélectionnée, il peut commencer l'opération en déplaçant le stylet à partir du point d'activation. Illustration extraite de [Tian 08].....	103

Chapitre 6

- Figure 1 : Modèle d'exploration des systèmes d'information. Illustration extraite de [Waterworth 91]..... 114
- Figure 2 : Structuration de l'exploration pour les systèmes de menus..... 115
- Figure 3 : Familiarité d'un utilisateur en fonction de la profondeur du menu, de son expertise et la précision de son but. Illustration réalisée à partir de [Norman 91]. 117
- Figure 4 : Arbre de commande simplifié du logiciel de statistiques « Stata ». 118
- Figure 5 : Deux règles d'arrêt dans une stratégie de lecture : la règle d'optimisation (gauche) et la règle de satisfaction (droite). Illustration extraite de [Norman 91]. 120
- Figure 6 : Les 3 types d'inspection. Inspection en série, inspection aléatoire sans remplacement, inspection aléatoire avec remplacement. Illustration extraite de [Norman 91]. 120
- Figure 7 : Processus de correspondance entre le nom de la commande recherchée et le libellé de l'item balayé. Illustration extraite de [Norman 91]. 121
- Figure 8 : A gauche, un menu hiérarchique sans prévisualisation. L'utilisateur peut sélectionner seulement les items du menu courant. Pour sélectionner les items du sous-menu, il doit explicitement demander l'ouverture du sous-menu. A droite, un menu hiérarchique avec prévisualisation. L'utilisateur peut sélectionner les items du sous-menu car le sous-menu a été ouvert automatiquement. 123
- Figure 9 : Exploration en largeur de 3 sous-menus avec un Compound Marking menu. Grâce à la prévisualisation, l'utilisateur peut visiter 3 sous-menus continûment sans clic souris. 123
- Figure 10 : Exploration en largeur de 3 sous-menus avec un Multi-Stroke menu (qui n'a pas la propriété de prévisualisation). L'utilisateur doit : 1) explicitement demander l'ouverture d'un sous menu, 2) le parcourir..... 123
- Figure 11 : A gauche, un menu hiérarchique dont le chemin parcouru est observable. Le chemin déjà parcouru dans la branche courante est affiché mais les items ne peuvent pas être sélectionnés. A droite, un menu hiérarchique qui permet d'accéder directement aux parents. 124
- Figure 12 : Outil « Snap shots » permet d'avoir un aperçu de page accessible à partir d'un lien hypertexte. Illustration extraite de [Snapshots]. 125
- Figure 13 : Les Compound Marking menus (CMM). A gauche, le mode novice. A droite le mode expert. Illustration extraite de [Zhao 04]. 126
- Figure 14 : Les Multi-Stroke menus (MSM). A gauche le mode novice, à droite, le mode expert. Illustration extraite de [Zhao 04]. 127
- Figure 15 : Fonctionnement d'un Wave menu à deux niveaux de profondeur. (état A) Le menu apparaît centré autour du curseur. En dessinant une marque en direction de l'item (transition 1), le premier niveau de la hiérarchie s'élargit pour laisser apparaître progressivement le sous-menu (état B). En appuyant sur le sous-menu, celui-ci prend le focus (Etat C). Avec une seconde marque (transition 2), l'utilisateur active l'item. Une confirmation rappelle pendant 300ms, l'item activé (Etat D)..... 128
- Figure 16 : Fonctionnement d'un Wave menu à trois niveaux. (état A) Le menu apparaît centré autour du curseur. En dessinant une marque en direction de l'item (transition 1), le premier niveau de la hiérarchie s'élargit pour laisser apparaître progressivement le sous-menu (état B). Avec une seconde marque (transition 2), le troisième niveau apparaît au centre et les deux premiers niveaux s'élargissent (état C). Cet effet rappelle la propagation des vagues. 129

Figure 17 : Parcours visuel important durant la recherche visuelle dans les CMM.	130
Figure 18 : Les gestes sont indépendants à l'échelle en mode novice. Pour sélectionner une commande, l'utilisateur peut relâcher le bouton avant d'avoir atteint l'item (b) ou après l'avoir dépassé (d). Pendant l'interaction, les menus parents sont inactifs (semi-transparents) pour ne pas être sélectionnés par erreur.	131
Figure 19 : Remontée dans la hiérarchie avec les Wave menus. L'utilisateur clique dans la zone centrale (a). Une rapide animation est jouée montrant le menu parent qui se rétracte (b).	131
Figure 20 : Surface d'entrée minimale pour utiliser un Wave menu quelque soit sa profondeur. L'utilisateur peut réaliser des marques superposées pour descendre progressivement dans la hiérarchie.	133
Figure 21 : Les Wave menus sont utilisables près des bords de l'écran en mode dégradé (la prévisualisation ou l'accès direct aux parents ne sont plus possibles).	134
Figure 22 : Représentations du Wave menu. (a) la représentation adoptée. (b) une représentation qui met en valeur la hiérarchie, mais qui est occulte fortement le contexte. (c) une représentation qui maximise le contexte visuel, mais qui ne permet pas de comprendre la hiérarchie.	135
Figure 23 : Wave menus inversés (WM2).	136
Figure 24 : Détour forcé dans les Wave menus inversés.	136
Figure 25 : L'utilisateur doit sélectionner les différents items « Paris » dans les 4 sous-menus « Europe ».	140
Figure 26 : Temps d'exécution en secondes en fonction des techniques et du nombre d'items dans les sous-menus.	143
Figure 27 : Taux d'erreur en fonction des techniques et du nombre d'items dans le menu.	143

Chapitre 7

Figure 1 : Différentes structures de menus. Pour le même nombre de commandes (256) et la même profondeur (4), Norman [Norman 88] étudie la variation de la largeur en fonction de la profondeur. Illustration extraite de [Norman 91].	153
Figure 2 : Distribution du nombre d'items (en %) dans les menus (1 ^{er} niveau) et les sous-menus (2 ^{ème} niveau). Par exemple, la configuration la plus répandue consiste en des sous-menus avec 1-4 commandes.	154
Figure 3 : Distribution du nombre d'items dans les groupements internes des menus (1 ^{er} niveau) et sous-menus (2 ^{ème} niveau).	155
Figure 4 : Largeur de la barre de menus et de deux sous-menus dans Microsoft Word.	155
Figure 5 : Zhao [Zhao 06] considère en plus de l'orientation, la position relative par rapport à une origine (le point d'activation du menu). Illustration extraite de [Zhao 06].	157
Figure 6 : Tous les gestes utilisés (32) dans les Zone menus [Zhao 06].	157
Figure 7 : Les 16 gestes retenus pour les Polygon menus [Zhao 06].	158
Figure 8 : Les 7 gestes du Flower menu pour l'orientation nord et leur temps d'exécution respectifs.	158
Figure 9 : L'arc (S) et la corde (l) de courbe entre les points A et B et le rayon R.	159

Figure 10 : Les 56 gestes des Flower menus. $7 = 2 * 4 - 1$ gestes pour chacune des 8 orientations.	160
Figure 11 : L'interface graphique de l'expérience. Le stimulus est représenté en haut de l'écran. Le cercle indique le début du geste. Similairement, un cercle au centre de l'écran indique à l'utilisateur l'endroit où il doit commencer son geste.	162
Figure 12 : Ensemble des gestes inclinés dans le sens anti-horaire pour tous les utilisateurs..	164
Figure 13 : A gauche, un Marking menu équilibré. A droite, un Marking menu prenant en compte les biais de l'utilisateur (secteur plus large sur les diagonales et centré sur les axes préférentiels).....	164
Figure 14 : les Flower menus.....	165
Figure 15 : Zone Menus (ZM) [Zhao 06]. Illustration détaillée dans la section 1 du chapitre 5..	166
Figure 16 : Les Polygon menus (PM) [Zhao 06]. Illustration détaillée dans la section 1 du chapitre 5.....	166
Figure 17 : Représentation compactée des Marking menus qui ne permet plus d'utiliser des séparateurs (en pontillés) pour les groupements internes.	167
Figure 18 : 4 types de gestes utilisés dans les Flower menus. a) b) c) les menus contiennent les mêmes items que le menu « File » de Microsoft Word. D) Le menu contient les mêmes items que le menu « Outils » dans Microsoft Word.	168
Figure 19 : a) Mode novice et b) Mode expert des Flower menus.	168
Figure 20 : Action en mode expert d'une commande de profondeur 3. L'utilisateur dessine 3 marques superposées (un geste courbe, un geste droit et une boucle).....	169
Figure 21 : Comparaison des groupements internes dans les Flower menus et dans les menus linéaires. Les commandes « New », « Open » et « Close » qui appartiennent au même groupement interne sont sur la même branche et partagent le même début de geste.....	169
Figure 22 : Un Flower menu hiérarchique utilisant le mécanisme du Wave menu.....	170
Figure 23 : Distribution du nombre de commandes dans les menus de 6 applications grand public.....	173
Figure 24 : Sélection d'une commande et contrôle d'une valeur par « crossing » sur les gestes droits. (a) l'utilisateur fait un geste en direction de la commande zoom. (b) Une fois, le seuil atteint, le menu disparaît (c). L'utilisateur contrôle alors la valeur du zoom (d-e).....	174
Figure 25 : Sélection d'une commande et contrôle de valeur par « fermeture » sur les gestes boucles. (a) l'utilisateur commence une boucle pour sélectionner le « contrôle de la précision du zoom ». (b) une fois, qu'il a fermé la boucle, le menu disparaît (c). L'utilisateur contrôle alors le niveau de précision de la commande zoom (d-e).	174
Figure 26 : Combinaison de commandes avec les Flower menus. En haut, l'utilisateur dessine deux marques séquentiellement pour créer un rectangle rouge (commande « Ovale » puis commande « rouge »). En bas, l'utilisateur combine ces deux commandes en un seul geste (1 marque au lieu de 2). Pour cela, il dessine la marque pour la forme ovale à l'envers qu'il concatène avec la marque pour la couleur rouge.	175
Figure 27 : le menu « file » de Microsoft Word 2003.....	179
Figure 28 : Les trois menus utilisés dans notre expérience : (a) Flower menu, (b) Polygon menu, (c) menu linéaire.....	179
Figure 29 : Performance de la mémorisation du mode expert des commandes pour les trois techniques.....	182

Figure 30 : Indirection entre l'orientation de l'item, de la marque et du décalage tangentiel dans les Polygon menus.....	185
Figure 31 : A gauche, deux gestes proches (même orientation) donnent accès à deux items éloignés spatialement. A droite, deux items proches spatialement ont des gestes très différents.	186
Figure 32 : A gauche, le Polygon menu. A droite, une nouvelle représentation du Polygon menu qui supprime une partie des indirections.....	186
Figure 33 : Indirection au début avec les Polygon menus. Indirection à la fin avec les Flower menus.....	187

Chapitre 8

Figure 1 : Contraintes associées à l'interaction au pouce sur dispositif mobile. Illustration extraite de [Roudaut 08].....	192
Figure 2 : ArchMenu [Huot 07]. Menu semi-circulaire qui s'active dans le coin en bas à droite de l'écran avec le pouce. Illustration extraite de [Huot 07].....	193
Figure 3 : ThumbMenu [Huot 07]. Illustration extraite de [Huot 07].....	193
Figure 4 : Mécanisme de rotation du ThumbMenu. A) l'utilisateur revient au centre du menu afin que b) le système soit pivoté et le sous-menu affiché dans le prolongement du pouce. Illustration extraite de [Huot 07].....	194
Figure 5 : Mode novice du Leaf menu : a) l'utilisateur presse une cible avec son pouce ; b) après un délai de 300ms, le menu apparaît. Une marque est associée à chaque item ; c) l'utilisateur relâche son doigt sur l'item désiré pour l'activer ; d) un retour visuel confirme l'item sélectionné et indique le geste à effectuer en mode expert.....	194
Figure 6 : Effet miroir du Leaf menu : la distance d'un item au point d'activation est invariante.	195
Figure 7 : Mode expert du Leaf menu : a) l'utilisateur presse une cible puis b) dessine directement la marque associée à l'item désirée. A la fin de son geste, un retour visuel apparaît pour confirmer l'item sélectionné.....	195
Figure 8 : Les différents raccourcis gestuels associés à l'item 6 ("Ajouter") de la Figure 6.....	196
Figure 9 : Positionnement d'un menu linéaire et ordonnancement de ses items : a) le point d'activation est près du bord haut de l'écran ; b) le point d'activation est près du bord bas de l'écran.....	197
Figure 10 : Exemple de trace associé à un Leaf menu hiérarchique avec 3 niveau de profondeur.	197
Figure 11 : Groupements internes dans les Leaf menus.....	198
Figure 12 : Les 4 raccourcis gestuels associés à l'item 6 permettant la sélection de cet item en mode expert quelque soit l'endroit où se situe le point d'activation.	199
Figure 13 : Sélection d'un item au pouce avec le Leaf menu sur un PDA-Phone.	200
Figure 14 : pourcentage de sélection, réussie en mode novice, ratée en mode novice, réussie en mode expert et ratée en mode expert en fonction de l'indice du bloc.....	201
Figure 15 : Pourcentage de sélections correctes en mode expert (SCE) en fonction des différents items et de l'indice du bloc.....	202
Figure 16 : Différentes parties de la main et les nom des principales articulations.	204
Figure 17 : 3 phalanges d'un doigt long. Illustration extraite de [Ouhaddi 98].....	204
Figure 18 : A gauche, les mouvements d'adduction et d'abduction. A droite, les mouvements d'extensions et de flexion. Illustration extraite de [Ouhaddi 98].....	205

Figure 19 : DataGlove. Illustration extraite de http://www.virtualtechnologiesltd.com/	206
Figure 20 : DigitalDesk. Illustration extraite de [Wellner 93].....	206
Figure 21 : Technologie de Jeff Han [Han 05] basé sur la réflexion totale contrariée (FTIR). Illustration extraite de [Han 05].....	207
Figure 22 : Lemur : écran LCD tactile permettant de capturer une dizaine de points de contact simultanément. Illustration extraite de [JazzMutant].....	207
Figure 23 : vocabulaire gestuel d'Apple. a) le « Pitch & Expand » utilisé pour zoomer un élément. b) le « Rotate » pour la rotation d'un objet. c) Le « Scroll », translation à deux doigts pour faire défiler un document. d) le « Three-finger Swipe » utilisé pour « jeter » un document. e) le « Double tap », appui simultané de deux doigts pour ouvrir un menu. Illustration extraite de [TrackpadGestures].....	208
Figure 24 : Extrait du brevet « Mutli-Touch Gesture Dictionary ». Illustration extraite de [PatentApple 07].	208
Figure 25 : « sweeping vertical » pour déplacer un groupe d'objets. Illustration extraite de [Wu 03].....	209
Figure 26 : A gauche, les cotés de la main délimite une zone de sélection. A droite, les mains positionnées en coin définissent une zone privée visible que de l'utilisateur. Illustrations extraites de [Wu 03].	209
Figure 27 : « Dual Finger X-Menu [Benko 06]. Le menu circulaire est activé dès que deux doigts sont posés sur la table. Illustration extraite de [Benko 06].....	209
Figure 28 : menu de Brandl. Un menu semi-circulaire qui s'active de préférence avec le majeur permettant à l'index de sélectionner un item. Illustration extraite de [Brandl 08].....	210
Figure 29 : La « Furniture Palette » [Wu 03]. L'utilisateur « double tap » avec le pouce pour activer le menu. La sélection d'un item donne accès à une Toolglass qui est manipulée avec l'index. La technique implique donc l'utilisation de deux doigts, le pouce au premier niveau de la hiérarchie et l'index au second niveau. Illustration extraite de [Wu 03].	210
Figure 30 : menu de Gäser : le menu est ouvert tant que toute la paume est en contact avec la table. L'orientation de la paume permet de sélectionner un élément. Il est activé après un délai. Pour activer les éléments plus profonds, il est nécessaire d'utiliser une autre main. Illustration extraite de [Gläser 07].....	211
Figure 31 : menu de Koike [Koike 02]. Pour activer le menu, l'utilisateur doit poser les deux mains sur l'écran et replier tous les doigts de la main gauche.....	211
Figure 32 : manipulation du MTM, a) l'utilisateur assis à une table horizontale, b) l'utilisateu debout à une table horizontale, c) debout à une table inclinée. Dans les cas b) et c) l'angle entre la main et l'avant bras est fermé et peut provoquer de la fatigue au niveau du poignet.	213
Figure 33 : Résultats de notre algorithme des cinq doigts sur un écran Stantum. Chaque point de contact a été affecté correctement à un doigt (T : thumb, I : index, M : middle, R : ring, L : litle) et la main droite (Right a été reconue).....	213
Figure 34 : Algorithme pour identifier le doigt associé à chaque point.....	214
Figure 35 : Deux niveaux de hiérarchie du Multi-Touch Menu.	215
Figure 36 : Multi-Touch Menu. L'utilisateur fait un mouvement circulaire avec le pouce pour explorer les différents sous-menus. Une fois, la commande désirée trouvée, elle est activée avec le doigt long correspondant.....	216

Figure 37 : Pointage séquentiel de deux cibles. En haut avec un seul doigt : l'utilisateur sélectionne les deux cibles séquentiellement. En bas avec deux doigts : l'utilisateur sélectionne les deux cibles de manière concomitante (léger parallélisme).....	217
Figure 38 : Pointage de deux cibles en parallèle.....	217
Figure 39 : Activation d'une commande en mode expert avec le talon comme type d'activateur. A gauche, la trace associé à la posture de la main. A droite, son interprétation, en utilisant le talon comme référentiel pour placer les différents éléments du menu.....	218
Figure 40 : Activation d'une commande en mode expert avec les cinq doigts comme type d'activateur. A gauche, l'utilisateur active le menu en posant les cinq doigts et fixe le référentiel. Au milieu, l'utilisateur pose le pouce et l'index de manière à activer la commande désirée. A droite, son interprétation.	218
Figure 41 : Groupements internes dans le MTM.....	219
Figure 42 : Utilisation des mouvements d'abduction de l'index et de l'auriculaire pour accéder à 4 nouveaux items.	220
Figure 43 : Décalage des libellés des commandes pour éviter les problèmes d'occultation de la main sur le menu.....	221
Figure 44 : Combinaison de commandes avec le MTM. Les deux commandes sont combinées pour mélanger les couleurs.	221

Publications personnelles

En relation avec les techniques de menus

- [Bailly 07a] Bailly, G., Lecolinet, E., and Nigay, L. (2007). Wave Menus : Improving the novice mode of Marking Menus. Conference Proceedings of INTERACT'07, International Conference on Human-Computer Interaction, Springer LNCS (Lecture Notes in Computer Science 4662, Rio de Janeiro, Brazil, 10-14 September, 2007. pp. 475-488.
- [Bailly 07b] Bailly, G., Lecolinet, E., Nigay, L. 2007. Quinze Ans de Recherche sur les Critères et Propriétés des Techniques de Menus. Conférence IHM 2007, 19^{ème} Conférence francophone sur l'Interaction Homme-Machine, Paris, 13-15 novembre 2007, ACM Press.
- [Bailly 08a] Bailly, G., Lecolinet, E., and Nigay, L. 2008. Flower menus: a new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. In Proceedings of the Working Conference on Advanced Visual interfaces (Napoli, Italy, May 28 - 30, 2008). AVI '08. ACM, New York, NY, pp. 15-22.
- [Bailly 08b] Bailly, G., Roudaut, A., Lecolinet, E., Nigay, L. 2008. Menus Leaf : Enrichir les menus lineaires par des gestes. Conférence IHM 2008, 20^{ème} Conférence francophone sur l'Interaction Homme-Machine, Metz 2-5 septembre 2008, ACM Press.
- [Bailly 08c] Bailly, G., Demeure, A., Lecolinet, E., Nigay, L. 2008. Multi-Touch Menu (MTM). Conférence IHM 2008, 20^{ème} Conférence francophone sur l'Interaction Homme-Machine, Metz 2-5 septembre 2008, ACM Press.

Autres publications

- [Bailly 05] Bailly, G., Nigay L. and Auber, D. 2005. 2M: un Espace de Conception pour l'Interaction Bi-Manuelle. Actes des 2^{èmes} Journées Francophone : Mobilité et Ubiquité, Ubimob'05, Grenoble, France, 31 mai-3 juin 2005, ACM Press, pp. 177-184.
- [Bailly 06] Bailly, G., Auber, D., and Nigay, L. 2006. From Visualization to Manipulation of RNA Secondary and Tertiary structures. Conference Proceedings of IV'06, Conference on Information Visualization, London, UK, 4-6 July 2006, pp. 107-116.
- [Bailly 06] Bailly, G., Nigay, L. and Auber, D. 2006. Navrna: Visualization – exploration – editing of RNA. Conference Proceedings of AVI'06, Advanced Visual Interfaces, Venezia, Italy, 23-26 May 2006, pp. 504-507.
- [Bailly 07c] Tahir, M., Bailly, G., Lecolinet, E. 2007. ARemote: A tangible interface for selecting TV channels. Conference Proceedings of the 17th Annual Conference on Artificial Reality and Telexistence,

ICAT'07, IEEE, Springer, Esbjerg, Denmark, 23-28 november 2007, ACM Press, pp. 298-299.

[Bailly 08d] Thierry-Mieg, N. and Bailly, G. 2008. Interpool : Interpreting Smart Pooling Results. *Bioinformatics*. 24(5) pp. 696-703.

[Bailly 08e] Tahir, M. , Bailly, G. and Lecolinet E. (2008). TactiMote: A Tactile Remote Control for Navigating in Long Lists. Conference Proceedings of the 10th International Conference on Multimodal interfaces, ICMI'08, Chania, Crete, Greece, October 20-22, 2008, ACM Press, pp. 298-299.

Bibliographie

- [Abowd 92] Abowd, G., Coutaz, J. and Nigay, L. 1992. Structuring the Space of Interactive System Properties. EHCI, Elsevier Science Pub, pp. 113-128.
- [Accot 97] Accot, J. and Zhai, S. 1997. Beyond Fitts' law: models for trajectory-based HCI tasks. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Atlanta, Georgia, United States, March 22 - 27, 1997). S. Pemberton, Ed. CHI '97. ACM, New York, NY, pp. 295-302.
- [Ahlström 05] Ahlström, D. 2005. Modeling and improving selection in cascading pull-down menus using Fitts' law, the steering law and force fields. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Portland, Oregon, USA, April 02 - 07, 2005). CHI '05. ACM, New York, NY, pp. 61-70.
- [Ahlström 06] Ahlström, D., Alexandrowicz, R., and Hitz, M. 2006. Improving menu interaction: a comparison of standard, force enhanced and jumping menus. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI '06. ACM, New York, NY, pp. 1067-1076.
- [Apitz 04] Apitz, G. and Guimbretière, F. 2004. CrossY: a crossing-based drawing application. In Proceedings of the 17th Annual ACM Symposium on User interface Software and Technology (Santa Fe, NM, USA, October 24 - 27, 2004). UIST '04. ACM, New York, NY, pp. 3-12.
- [Appert 04] Appert, C., Beaudouin-Lafon, M., and Mackay., W. 2004. Context matters: Evaluating interaction techniques with the CIS model. In Proceedings People and Computers XVIII - Design for Life -, HCI 2004, Leeds, UK. Springer Verlag, September 2004, pp. 279-295.
- [Appert 06a] Appert, C. and Fekete, J. 2006. OrthoZoom scroller: 1D multi-scale navigation. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI '06. ACM, New York, NY, pp. 21-30.
- [Appert 06b] Appert, C. and Fekete, J.D. 2006. ControlTree: Navigating and Selecting in a Large Tree. Demo adjunct Proceedings ACM symposium on User Interface Software and Technology, UIST 2006, Montreux, Suisse, pp. 47-48.
- [Appert 06c] Appert, C. and Beaudouin-Lafon, M. 2006. SMCanvas: augmenter la boîte à outils Java Swing pour prototyper des techniques

d'interaction avancées. In Proceedings of the 18th international Conference of the Association Francophone D'interaction Homme-Machine (Montreal, Canada, April 18 - 21, 2006). IHM '06, vol. 133. ACM, New York, NY, pp. 99-106.

- [Appert 07] Appert, C., 2007. Modélisation, Evaluation et Génération de Techniques d'Interaction. Université Paris-Sud. Mai 2007, Orsay, France.
- [Appert 08] Appert, C., Chapuis, O., and Beaudouin-Lafon, M. 2008. Evaluation of pointing performance on screen edges. In Proceedings of the Working Conference on Advanced Visual interfaces (Napoli, Italy, May 28 - 30, 2008). AVI '08. ACM, New York, NY, 119-126.
- [Appert 09] Appert, C., Zhai, S. 2009. Using Strokes as Command Shortcuts: Cognitive Benefits and Toolkit Support. In Proc. ACM Conference on Human Factors in Computing Systems, CHI 2009, Boston, USA. To appear.
- [Atkinson 68] Atkinson, R.C. and Shiffrin, R.M. 1968. Human memory: A proposed system and its control processes. In K.W. Spence and J.T. Spence (Eds.), The psychology of Learning and motivation, vol. 8. London: Academic Press.
- [Baddeley 78] Daddely, A. D. and Longman, D. J. A. 1978. The influence of length and frequency of training sessions on rate of Learning to type. Ergonomics, 21, pp. 627 635.
- [Bailly 07a] Bailly, G., Lecolinet, E., and Nigay, L. 2007. Wave Menus : Improving the novice mode of Marking Menus. Conference Proceedings of INTERACT'07, International Conference on Human-Computer Interaction, Springer LNCS (Lecture Notes in Computer Science 4662, Rio de Janeiro, Brazil, 10-14 September, 2007. pp. 475-488.
- [Bailly 07b] Bailly,G., Lecolinet, E., Nigay, L. 2007. Quinze Ans de Recherche sur les Critères et Propriétés des Techniques de Menus. Conférence IHM 2007, 19^{ème} Conférence francophone sur l'Interaction Homme-Machine, Paris, 13-15 novembre 2007, ACM Press.
- [Bailly 08a] Bailly, G., Lecolinet, E., and Nigay, L. 2008. Flower menus: a new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. In Proceedings of the Working Conference on Advanced Visual interfaces (Napoli, Italy, May 28 - 30, 2008). AVI '08. ACM, New York, NY, pp. 15-22.
- [Bailly 08b] Bailly, G., Roudaut, A., Lecolinet, E., Nigay, L. 2008. Menus Leaf : Enrichir les menus lineaires par des gestes. Conférence IHM 2008, 20^{ème} Conférence francophone sur l'Interaction Homme-Machine, Metz 2-5 septembre 2008, ACM Press.
- [Bailly 08c] Bailly, G., Demeure, A., Lecolinet, E., Nigay, L. 2008. Multi-Touch Menu (MTM). Conférence IHM 2008, 20^{ème} Conférence francophone

sur l'Interaction Homme-Machine, Metz 2-5 septembre 2008, ACM Press.

- [Balbo 94] Balbo, S. 1994. Evaluation ergonomique des interfaces utilisateur : un pas vers l'automatisation. Thèse de doctorat en informatique préparée au Laboratoire de Génie Informatique (IMAG), Université Joseph Fourier, 287 pages.
- [Balakrishnan 98] Balakrishnan, R. and Patel, P. 1998. The PadMouse: facilitating selection and spatial positioning for the non-dominant hand. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Los Angeles, California, United States, April 18 - 23, 1998). C. Karat, A. Lund, J. Coutaz, and J. Karat, Eds. Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co., New York, NY, pp. 9-16.
- [Balentine 99] Balentine, B. and Morgan D. P. 1999. How to build a speech recognition application: a style guide for telephony dialogues. San Ramon : Enterprise Integration Group.
- [Barfield 93] Barfield, L. 1993 *The User Interface Concepts & Design*. Addison Wesley Publishing.
- [Barnard 87] Barnard, P. 1987. Cognitive Ressources and the Learning of Computer Dialogs. In *interfacing Thought, Cognitive aspects of Human Computer Interaction*, J.M. Carroll Ed., MIT Press Publ., pp. 112-158.
- [Bastien 93] Bastien, J.M.C., Scapin, D. 1993 Ergonomic Criteria for the évaluation of Human-Computer interfaces. Institut National de recherche en informatique et en automatique.
- [Bau 08] Bau, O. and Mackay, W. E. 2008. OctoPocus: a dynamic guide for learning gesture-based command sets. In Proceedings of the 21st Annual ACM Symposium on User interface Software and Technology (Monterey, CA, USA, October 19 - 22, 2008). UIST '08. ACM, New York, NY, 37-46.
- [Beaudouin-Lafon 97] Beaudouin-Lafon, M. 1997. Interaction instrumentale : de la manipulation directe à la réalité augmentée. Neuvième Journée sur l'Ingénierie des Interfaces Homme-Machine, IHM'97, Poitiers, septembre 1997, Cépaduès-Editions.
- [Beaudouin-Lafon 00] Beaudouin-Lafon, M. 2000. Instrumental interaction: an interaction model for designing post-WIMP user interfaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 446-453.
- [Bederson 95] Bederson, B. B. and Hollan, J. D. 1995. Pad++: a zoomable graphical interface system. In Conference Companion on Human Factors in Computing Systems (Denver, Colorado, United States, May 07 - 11,

- 1995). I. Katz, R. Mack, and L. Marks, Eds. CHI '95. ACM, New York, NY, pp. 23-24.
- [Bederson 00] Bederson, B. B. 2000. Fisheye menus. In Proceedings of the 13th Annual ACM Symposium on User interface Software and Technology (San Diego, California, United States, November 06 - 08, 2000). UIST '00. ACM, New York, NY, pp. 217-225.
- [Benyon 97] Benyon, D. and Höök, K. 1997. Navigation in Information Spaces: supporting the individual. In Proceedings of the IFIP Tc13 international Conference on Human-Computer interaction (July 14 - 18, 1997). S. Howard, J. Hammond, and G. Lindgaard, Eds. IFIP Conference Proceedings, vol. 96. Chapman & Hall Ltd., London, UK, pp. 39-46.
- [Bernard 02] Bernard, M. L. 2002. Examining the effects of hypertext shape on User Performance. Usability News, 4.2.
- [Bersen 96] Bersen, N.O. 1996. A taxonomy of input modalities. Amodeus Project Deliverable TM/WP22.
- [Bier 93] Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. 1993. Toolglass and magic lenses: the see-through interface. In Proceedings of the 20th Annual Conference on Computer Graphics and interactive Techniques (Anaheim, CA, August 02 - 06, 0093). SIGGRAPH '93. ACM, New York, NY, pp. 73-80.
- [Blanch 04a] Blanch, R. 2004. Pointage sémantique et distracteurs, la dynamique du pointage a la rescousse. Dans les Actes des secondes rencontres jeunes chercheurs en interaction homme-machine (RJC-IHM 2004). Octobre 2004.
- [Blanch 04b] Blanch, R., Guiard, Y., and Beaudouin-Lafon, M. 2004. Semantic pointing: improving target acquisition with control-display ratio adaptation. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, pp. 519-526.
- [Blanch 05] Blanch, R. 2005. Architecture logicielle et outils pour les interfaces hommes-machines graphiques avancées. Thèse de doctorat, Université Paris-Sud XI Orsay.
- [Bowes 03] Bowes, J., Dearman, D., Perkins, R. 2003 Transparency for item highlighting. Poster Presentation at Graphics Interface 2003, Halifax, Nova Scotia.
- [Brandl 08] Brandl, P., Forlines, C., Wigdor, D., Haller, M., and Shen, C. 2008. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In Proceedings of the Working Conference on Advanced Visual interfaces (Napoli, Italy, May 28 - 30, 2008). AVI '08. ACM, New York, NY, pp. 154-161.

- [Buisson 03] Buisson, M., Sallagoïty, I., Athènes, S., and Mertz, C. 2003. From human movement analysis to interface design: applications to writing and gesture based user interface. In Proceedings of the 15th French-Speaking Conference on Human-Computer interaction on 15eme Conference Francophone Sur L'interaction Homme-Machine (Caen, France, November 25 - 28, 2003). T. Baudel, Ed. IHM 2003, vol. 51. ACM, New York, NY, pp. 224-227.
- [Buxton 89] Buxton, W. 1989. Introduction to this special issue on nonspeech audio. *Journal of Human-Computer Interaction*, 4, pp. 1-9.
- [Buxton 95] Buxton, W. A. 1995. Chunking and phrasing and the design of human-computer dialogues. In *Human-Computer interaction: Toward the Year 2000*, R. M. Baecker, J. Grudin, W. A. Buxton, and S. Greenberg, Eds. Morgan Kaufmann Publishers, San Francisco, CA, pp. 494-499.
- [Buxton 05] Buxton, W. 2005. Experience Design vs. Interface Design. *Rotman Magazine, The Alumni Magaizne of the Rotman School of Management*, Winter 2005, pp. 47-49.
- [Cadoz 94] Cadoz, C. 1994. Lee geste, canal de communication homme/machine: la communication instrumentale. *Technique et Science de l'information*, 13(1): pp. 31-61.
- [Callahan 88] Callahan, J., Hopkins, D., Weiser, M., and Shneiderman, B. 1988. An empirical comparison of pie vs. linear menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Washington, D.C., United States, May 15 - 19, 1988). J. J. O'Hare, Ed. CHI '88. ACM, New York, NY, pp. 95-100.
- [Calvary 02] Calvary, G. 2002. Ingénierie de l'interaction homme machine : rétrospective et perspectives. In *Traité des Sciences et Techniques de l'Information - Interaction homme-machine et recherche d'information*, C. Paganelli Ed., Hermès, pp. 19-63.
- [Cance 06] Cance, J., Collomb, M., Hascoët, M., Accelerating Object-command transitions with pie menus. *Proceeding of 3rd international conference on Enactive Interfaces, Enactive'06*, pp. 109-110.
- [Cao 07] Cao, X. and Zhai, S. 2007. Modeling human performance of pen stroke gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, pp. 1495-1504.
- [Card 83] Card, S., Moran, T., Newell, A. 1983. *The psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates.
- [Card 90] Card, S. K., Mackinlay, J. D., and Robertson, G. G. 1990. The design space of input devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Empowering People*

- (Seattle, Washington, United States, April 01 - 05, 1990). J. C. Chew and J. Whiteside, Eds. CHI '90. ACM, New York, NY, 117-124.
- [Card 99] S. K. Card, J. D. Mackinlay, and B. Shneiderman, Eds. 1999. Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann Publishers Inc.
- [Carmel 92] Carmel, E., Crawford, S., and Hsischun, C. 1992. Browsing in hypertext : A cognitive study. IEEE Transactions on System Man, and Cybernetics. 22, 5, pp. 865-883.
- [Catledge 95] Catledge, L. D. and Pitkow, J. E. 1995. Characterizing browsing strategies in the World-Wide Web. Comput. Netw. ISDN Syst. 27, 6 (Apr. 1995), pp. 1065-1073.
- [Chigona 02] Chigona, W. and Strothotte, T. 2002. Contextualized preview of image map links. In Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia (College Park, Maryland, USA, June 11 - 15, 2002). J. Blustein, Ed. HYPERTEXT '02. ACM, New York, NY, 150-158.
- [Chuah 96] Chuah, M. C. and Roth, S. F. 1996. On the semantics of interactive visualizations. In Proceedings of the 1996 IEEE Symposium on information Visualization (INFOVIS '96) (October 28 - 29, 1996). INFOVIS. IEEE Computer Society, Washington, DC, 29.
- [Cockburn 06] Cockburn, A. and Gin, A. 2006. Faster cascading menu selections with enlarged activation areas. In Proceedings of Graphics interface 2006 (Quebec, Canada, June 07 - 09, 2006). ACM International Conference Proceeding Series, vol. 137. Canadian Information Processing Society, Toronto, Ont., Canada, pp. 65-71.
- [Cockburn 07] Cockburn, A., Kristensson, P. O., Alexander, J., and Zhai, S. 2007. Hard lessons: effort-inducing interfaces benefit spatial learning. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, pp. 1571-1580.
- [Cockburn 07] Cockburn, A., Gutwin, C., and Greenberg, S. 2007. A predictive model of menu performance. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, pp. 627-636.
- [Cohen 04] Cohen, M. H., Giangola, J. P., and Balogh, J. 2004. Voice User Interface Design . Addison-Wesley Professional.
- [Commarford 08] Commarford, P. M., Lewis, J.R., and Smither, J.A., and Gentzler M.D. 2008 A comparison of broad versus deep auditory menu structures. Human Factors : The Journal of the Human Factors and Ergonomics Society 50 (February), pp. 77-89.

- [Coolpreview] <http://www.coolpreviews.com/>
- [Coutaz 94] Coutaz, J., Nigay, L. 1994. Les propriétés CARE dans les interfaces multimodales. Actes d'IHM'94, Lille, pp. 7-14.
- [Dachselt 07] Dachselt, R., Hübner, A. 2007 Virtual Environments: Three-dimensional menus: A survey and taxonomy. Comput. Graph. 31, 1 (Jan. 2007), pp. 53-65.
- [PatentApple 07] <http://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=%2Fnethtml%2FPTO%2Fsarch-adv.html&r=1&p=1&f=G&l=50&d=PG01&S1=20070177804&OS=20070177804&RS=20070177804>
- [PatentApple 08] <http://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=%2Fnethtml%2FPTO%2Fsrchnum.html&r=1&f=G&l=50&s1=%2220080163130%22.PGNR.&OS=DN/20080163130&RS=DN/20080163130>
- [Dietz 01] Dietz, P. and Leigh, D. 2001. DiamondTouch: a multi-user touch technology. In Proceedings of the 14th Annual ACM Symposium on User interface Software and Technology (Orlando, Florida, November 11 - 14, 2001). UIST '01. ACM, New York, NY, pp. 219-226.
- [Dillon 90] Dillon, R. F., Edey, J. D., and Tombaugh, J. W. 1990. Measuring the true cost of command selection: techniques and results. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Empowering People (Seattle, Washington, United States, April 01 - 05, 1990). J. C. Chew and J. Whiteside, Eds. CHI '90. ACM, New York, NY, pp. 19-26.
- [Dix 95] Dix, A. 1995. Accelerators and tollbars: Learning from the menu. HCI'95 adjunct proceedings (eds.) G. Allen, J. Wilkinson, P. Wright (School of Computing & Mathematics, University of Huddersfield), pp. 138-143.
- [Dix 03] Dix, A., Finlay, J. E., Abowd, G. D. and Beale, R. 2003. Human-Computer Interaction. Prentice hall ; 3 edition (December 20, 2003), 834 pages.
- [Dragicevic 04] Dragicevic, P. 2004. Un modèle d'interaction en entrée pour des systèmes interactifs multi-dispositifs hautement configurables. Thèse de doctorat, Université de Nantes.
- [Dubois 01] Dubois, E., "Chirurgie Augmentée :un Cas de Réalité Augmentée; Conception et Réalisation Centrée sur l'Utilisateur" , Thèse de Doctorat de l'Université Joseph Fourier , Grenoble , juillet 2001, 275 pages.

- [Fitts 54] Fitts, P.M. The Information Capacity of The Human Mote; System in Controlling The Amplitude of Movement. *Journal of Experimental Psychology*, 1954, 47, pp. 381-391.
- [Fitzmaurice 03] Fitzmaurice, G., Khan, A., Pieké, R., Buxton, B., and Kurtenbach, G. 2003. Tracking menus. In *Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology* (Vancouver, Canada, November 02 - 05, 2003). *UIST '03*. ACM, New York, NY, pp. 71-79.
- [Findlater 04] Findlater, L. and McGrenere, J. 2004. A comparison of static, adaptive, and adaptable menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria, April 24 - 29, 2004). *CHI '04*. ACM, New York, NY, pp. 89-96.
- [Fitzmaurice 08] Fitzmaurice, G., Matejka, J., Khan, A., Glueck, M., and Kurtenbach, G. 2008. PieCursor: merging pointing and command selection for rapid in-place tool switching. In *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy, April 05 - 10, 2008). *CHI '08*. ACM, New York, NY, pp. 1361-1370.
- [Foley 90] Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. 1990 *Computer Graphics: Principles and Practice* (2nd Ed.). Addison-Wesley Longman Publishing Co., Inc.
- [Forsey 84] Forsey, D., 1984 *Harmoy in Transposition: A Tocatta for Vax and M68000*. M.Sc. Thesis, University of Waterloo, Canada
- [Forsström 06] Forsström, E. 2006. *Marking menus for 3D Desktop Haptic Applications*. Master's thesis, Umea University, Sweden.
- [Friedlander 98] Friedlander, N., Schlueter, K., and Mantei, M. 1998. Bullseye! when Fitts' law doesn't fit. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Los Angeles, California, United States, April 18 - 23, 1998). C. Karat, A. Lund, J. Coutaz, and J. Karat, Eds. *Conference on Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, 257-264.
- [Furnas 86] Furnas, G. W. 1986. Generalized fisheye views. *SIGCHI Bull.* 17, 4 (Apr. 1986), pp. 16-23.
- [Furnas 97] Furnas, G. W. and Jul, S. 1997. Workshop on navigation in electronic worlds. In *CHI '97 Extended Abstracts on Human Factors in Computing Systems: Looking To the Future* (Atlanta, Georgia, March 22 - 27, 1997). *CHI '97*. ACM, New York, NY, pp. 230-230.
- [Furnas 97b] Furnas, G. W. 1997. Effective view navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, United States, March 22 - 27, 1997). S. Pemberton, Ed. *CHI '97*. ACM, New York, NY, pp. 367-374.

- [Gibson 79] Gibson, J.J. 1979. The ecological approach to visual perception. Houghton Mifflin, New York.
- [Gibson 82] Gibson, J.J. 1982. Reasons for realism : Selected essays of James J. Gibson. Reed, E. & Jones, R. (Ed.), Hillsdale NJ :Lawerence Erl aum.
- [Gläser 07] <http://fr.youtube.com/watch?v=JmHNR9EH1iU>
- [Grossman 05] Grossman, T. and Balakrishnan, R. 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Portland, Oregon, USA, April 02 - 07, 2005). CHI '05. ACM, New York, NY, pp. 281-290.
- [Grossman 07] Grossman, T., Dragicevic, P., and Balakrishnan, R. 2007. Strategies for accelerating on-line learning of hotkeys. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, pp. 1591-1600.
- [Grudin 92] Grudin, J. 1992. Utility and usability : Resarch issues and development contexts. In Interacting with Computers, 4, 2, (August), pp. 209-217.
- [Guiard 04] Guiard, Y., Blanch, R., and Beaudouin-Lafon, M. 2004. Object pointing: a complement to bitmap pointing in GUIs. In Proceedings of Graphics interface 2004 (London, Ontario, Canada, May 17 - 19, 2004). ACM International Conference Proceeding Series, vol. 62. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, pp. 9-16.
- [Guimbretiere 00] Guimbreti re, F. and Winograd, T. 2000. FlowMenu: combining command, text, and data entry. In Proceedings of the 13th Annual ACM Symposium on User interface Software and Technology (San Diego, California, United States, November 06 - 08, 2000). UIST '00. ACM, New York, NY, pp. 213-216.
- [Guimbreti re 05] Guimbreti re, F., Martin, A., and Winograd, T. 2005. Benefits of merging command selection and direct manipulation. ACM Trans. Comput.-Hum. Interact. 12, 3 (Sep. 2005), pp. 460-476.
- [Guimbreti re 07] Guimbreti re, F., Dixon, M., and Hinckley, K. 2007. ExperiScope: an analysis tool for interaction data. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, pp. 1333-1342.
- [Han 05] Han, J. Y. 2005. Low-cost multi-touch sensing through frustrated total internal reflection. In Proceedings of the 18th Annual ACM Symposium on User interface Software and Technology (Seattle,

- WA, USA, October 23 - 26, 2005). *UIST '05*. ACM, New York, NY, pp. 115-118.
- [Harrison 96] Harrison, B. L. and Vicente, K. J. 1996. An experimental evaluation of transparent menu usage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Common Ground* (Vancouver, British Columbia, Canada, April 13 - 18, 1996). M. J. Tauber, Ed. *CHI '96*. ACM, New York, NY, pp. 391-398.
- [Helander 97] Helander, M.G., Landauer, T. K., and Prabhu, P. V., Eds. 1997 *Handbook of Human-Computer Interaction*. 2nd. Elsevier Science Inc.
- [Hick 52] Hick, W., 1952. On the rate of gain of information. *Journal of Experimental Psychology*, Vol. 4, pp. 11-36.
- [Hinckley 05] Hinckley, K., Baudisch, P., Ramos, G., and Guimbretiere, F. 2005. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA, April 02 - 07, 2005). *CHI '05*. ACM, New York, NY, pp. 451-460.
- [Hinckley 99] Hinckley, K. and Sinclair, M. 1999. Touch-sensing input devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI Is the Limit* (Pittsburgh, Pennsylvania, United States, May 15 - 20, 1999). *CHI '99*. ACM, New York, NY, pp. 223-230.
- [Hinckley 06] Hinckley, K., Guimbretiere, F., Baudisch, P., Sarin, R., Agrawala, M., and Cutrell, E. 2006. The springboard: multiple modes in one spring-loaded control. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. *CHI '06*. ACM, New York, NY, pp. 181-190.
- [Häger-Ross 00] Häger-Ross, C., Schiebr, M. 2000. Quantifying the Independence of Human Finger Movements: Comparisons of Digits, Hands, and Movement Frequencies. *The Journal of Neuroscience*, November 15, 2000, 20(22), pp. 8542-8550.
- [Helson 33] Helson, H. 1933. The fundamental propositions of gestalt psychology. *Psychology Review*, 40, pp. 13-31.
- [Howes 94] Howes, A. 1994. A model of the acquisition of menu knowledge by exploration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Celebrating interdependence* (Boston, Massachusetts, United States, April 24 - 28, 1994). B. Adelson, S. Dumais, and J. Olson, Eds. *CHI '94*. ACM, New York, NY, pp. 445-451.
- [humanf. 03] <http://www.humanfactors.com/downloads/apr03.asp>.

- [humanf. 04] <http://www.humanfactors.com/downloads/jul04.asp>.
- [Huot 07] Huot, S. and Lecolinet, E. 2007. Archmenu et ThumbMenu : Contrôler son dispositif mobile « sur le pouce ». 19^{ème} Conférence francophone sur l'Interaction Homme-Machine (IHM'07), Paris, France, November 2007.
- [Huot 08] Huot, S., Nancel, M. and Beaudouin-Lafon, M. 2008. Push Menu : Extending Marking menus for Pressure-Enabled Input Services. Rapport de recherche n°1502.
- [Hyman 53] Hyman, R. 1953. Stimulus information as a déterminant of réaction time. *Journal of Experimental Psychology*. Vol. 45, pp. 188-196.
- [imuseum] <http://imuseum.lri.fr/>
- [Intelegen] http://web-us.com/memory/improving_memory.htm
- [iPhone] <http://www.apple.com/fr/iphone/>
- [ISO 9241-14] ISO. ISO/DIS 9241-14 *Ergonomic requirements for office work with visual display terminals (VDTs) –Part 14: Menu dialogues*. International Organization for Standardization, 1998.
- [Isokoski 02] Isokoski, P. and Käksi, M. 2002. Comparison of two touchpad-based methods for numeric entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves* (Minneapolis, Minnesota, USA, April 20 - 25, 2002). CHI '02. ACM, New York, NY, 25-32.
- [Jacko 96] Jacko, J.A., and Slavendy, G. 1996. Hierarchical Menu Design :breadth, depth and task complexity. *Perceptual and Motor skills*, 82, pp. 1187-1201.
- [Jakobsen 07] Jakobsen, M. R. and Hornæk, K. 2007. Transient visualizations. In *Proceedings of the 2007 Conference of the Computer-Human interaction Special interest Group (Chisig) of Australia on Computer-Human interaction: Design: Activities, Artifacts and Environments* (Adelaide, Australia, November 28 - 30, 2007). OZCHI '07, vol. 251. ACM, New York, NY, pp. 69-76.
- [Jackoby 92] Jacoby, R.H. and Ellis, S.R. 1992. Using virtual menus in a virtual environment. *Visual Data Interpretation*. V1668 i1, pp. 39-48.
- [JazzMutant] http://www.jazzmutant.com/lemur_overview.php
- [John 96] John, B.E., Kieras, D.E. 1996. The goms for user interface design and évaluation : which technique ? *ACM trans. Computer.Hum. Interact.*, 3(4), pp. 287-319.
- [Johnson 91] Johnson, B. and Shneiderman, B. 1991. Treemaps :A Space-filling Approach to the Visualization of Hierarchical Information. *IEEE Visualization'91*. San Diego, CA, pp. 284-291.

- [Jul 97] Jul, S. and Furnas, G. W. 1997. Navigation in electronic worlds: a CHI 97 workshop. SIGCHI Bull. 29, 4 (Oct. 1997), pp. 44-49.
- [Karlson 07] Karlson, A. K. and Bederson, B. B. ThumbSpace: generalized one-handed input for touchscreen-based mobile devices. In Proc. INTERACT 2007, Springer (2007), pp. 324-338.
- [Kieger 84] Kiger, J. I. 1984. The depth/breadth trade-off in the design of menu-driven user interfaces. Int. J. Man-Mach. Stud. 20, 2 (Mar. 1984), pp. 201-213.
- [Koike] Koike, H., Xinlei, C., Nakanishi, Y., Oka, K., and Sato, Y. 2002. Two-handed drawing on augmented desk. In CHI '02 Extended Abstracts on Human Factors in Computing Systems (Minneapolis, Minnesota, USA, April 20 - 25, 2002). CHI '02. ACM, New York, NY, pp. 760-761.
- [Krisler 08] Krisler, B. and Alterman, R. 2008. Training towards mastery: overcoming the active user paradox. In Proceedings of the 5th Nordic Conference on Human-Computer interaction: Building Bridges (Lund, Sweden, October 20 - 22, 2008). NordiCHI '08, vol. 358. ACM, New York, NY, pp. 239-248.
- [Kurtenbach 91] Kurtenbach, G. and Buxton, W. 1991. Issues in combining marking and direct manipulation techniques. In Proceedings of the 4th Annual ACM Symposium on User interface Software and Technology (Hilton Head, South Carolina, United States, November 11 - 13, 1991). UIST '91. ACM, New York, NY, 137-144.
- [Kurtenbach 93a] Kurtenbach, G. and Buxton, W. 1993. The limits of expert performance using hierarchic marking menus. In Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems. ACM CHI '93. pp. 482-487.
- [Kurtenbach 93b] Kurtenbach, G., 1993. The Design and Evaluation of Marking Menus. University of Toronto, 1993, Ph. D., Thesis, Dept. Of Computer Science.
- [Kurtenbach 94] Kurtenbach, G. and Buxton, W. 1994. User learning and performance with marking menus. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Celebrating interdependence (Boston, Massachusetts, United States, April 24 - 28, 1994). B. Adelson, S. Dumais, and J. Olson, Eds. CHI '94. ACM, New York, NY, pp. 258-264.
- [Kurtenbach 97] Kurtenbach 97. Display and control of menus with radial and linear portions. Filed April 1, 1997, patent #5,926,178.
- [Kurtenbach 97b] Kurtenbach, G., Fitzmaurice, G., Baudel, T., and Buxton, B. 1997. The design of a GUI paradigm based on tablets, two-hands, and transparency. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Atlanta, Georgia, United States, March

22 - 27, 1997). S. Pemberton, Ed. CHI '97. ACM, New York, NY, pp. 35-42.

- [Kurtenbach 99] Kurtenbach, G., Fitzmaurice, G. W., Owen, R. N., and Baudel, T. 1999. The Hotbox: efficient access to a large number of menu-items. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI Is the Limit (Pittsburgh, Pennsylvania, United States, May 15 - 20, 1999). CHI '99. ACM, New York, NY, pp. 231-237.
- [Lai 05] Lai, J.T., Anderson, R., Li, Y. 2005. A chinese Input Technique Using Pressure-Sensitive Marking menus. Proc. UIST'05.
- [Lamping 95] Lamping, J., Rao, R., and Pirolli, P. 1995. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Denver, Colorado, United States, May 07 - 11, 1995). I. R. Katz, R. Mack, L. Marks, M. B. Rosson, and J. Nielsen, Eds. Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co., New York, NY, pp. 401-408.
- [Landauer 85] Landauer, T. K. and Nachbar, D. W. 1985. Selection from alphabetic and numeric menu trees using a touch screen: breadth, depth, and width. SIGCHI Bull. 16, 4 (Apr. 1985), pp. 73-78.
- [Lane 05] Lane, D. M., Napier, H. A., Pers, S. C., Sandor, A. 2005. Hidden costs of Graphical User Interfaces : Failure to Make the Transition from Menus and Icon Toolbars to Keyboard Shortcuts. International Journal of Human-Computer Interaction, Vol. 16 (2), pp. 133-144.
- [Larson 98] Larson, K. and Czerwinski, M. 1998. Web page design: implications of memory, structure and scent for information retrieval. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Los Angeles, California, United States, April 18 - 23, 1998). C. Karat, A. Lund, J. Coutaz, and J. Karat, Eds. Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co., New York, NY, pp. 25-32.
- [Law 08] Law, E., Roto, V., Vermeeren, A. P., Kort, J., and Hassenzahl, M. 2008. Towards a shared definition of user experience. In CHI '08 Extended Abstracts on Human Factors in Computing Systems (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, pp. 2395-2398.
- [Lecolinet 03] Lecolinet, E. 2003. A molecular architecture for creating advanced GUIs. In Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, pp. 135-144.
- [Lecolinet 06] Lecolinet, E. and Nguyen, D. 2006. Représentation focus+contexte de listes hiérarchiques zoomables. In Proceedings of the 18th international Conference of the Association Francophone

- D'interaction Homme-Machine (Montreal, Canada, April 18 - 21, 2006). IHM '06, vol. 133. ACM, New York, NY, pp. 195-198.
- [Lee 93] Lee, E., Raymond, D. R. 1993. Menu-Driven Systems. Encyclopedia of Microcomputers, Vol. 11, pp. 101-127.
- [Lee 04] Lee, D. S., Yoon, W. C. 2004. Quantitative results assessing design issues of selection support menus. International Journal of Industrial Ergonomics, 33, pp. 41-52.
- [Lee 06] Lee, B., Plaisant, C., Parr, C. S., Fekete, J., and Henry, N. 2006. Task taxonomy for graph visualization. In Proceedings of the 2006 AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods For information Visualization (Venice, Italy, May 23 - 23, 2006). BELIV '06. ACM, New York, NY, pp. 1-5.
- [Lin 00] Lin, J., Wu, Y., and Huang, T. S. 2000. Modeling the constraints of human hand motion. In Proceedings of the Workshop on Human Motion (Humo'00) (December 07 - 08, 2000). HUMO. IEEE Computer Society, Washington, DC, 121.
- [Lindgaard 06] Lindgaard, G., Deudek, C., Fernandes, G., Brown, J. 2006. Attention web designers : You have 50 milliseconds to make a good first impression !. Behaviour & Information Technology, 25, pp. 115-126.
- [Lindgaard 07] Lindgaard, G. 2007. Aesthetics, Visual appeal, Usability and User Satisfaction : What Do the User's Eyes Tell the User's Brain ? In The Australian Journal of Emerging Technologies and Society, Vol. 5, No. 1, pp. 1-14.
- [Linton 99] Linton, F., Joy, D., and Schaefer, H. 1999. Building user and expert models by long-term observation of application usage. In Proceedings of the Seventh international Conference on User Modeling (Banff, Canada). J. Kay, Ed. Springer-Verlag New York, Secaucus, NJ, pp. 129-138.
- [Logickeyboard] <http://www.logickeyboard.com/>
- [Mackay 02] Mackay, W.E. 2002. Which Interaction Technique Works When? Floating Palettes, Marking Menus and Toolglasses support different task strategies. Proc. ACM Conference on Advanced Visual Interfaces (AVI'02), pp. 203-208.
- [Mackay 07] Mackay, W. E., Appert, C., Beaudouin-Lafon, M., Chapuis, O., Du, Y., Fekete, J., and Guiard, Y. 2007. Touchstone: exploratory design of experiments. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, pp. 1425-1434.
- [Marcus 94] Marcus, A., Silonich, N., Thompson, L. 1994. The Cross-Gui Handbook: For Multiplatform User Interface Design. Addison-Wesley Professional.

- [Marics 97] Marics, M.A., and Engelbeck, G. 1997. Designing voice menu applications for telephones. In Handbook of Human-Computer Interaction, pp. 1085-1102. Amsterdam : Elsevier.
- [Martin 07] Martin, B., Isokoski, P. 2007. Trackmouse Trackball in Pie Menu Use : Data on Accuracy. Conference Proceedings of IHM'07, Interaction Homme Machine, Paris, France, ACM Press, pp. 127-130.
- [Matsushita 97] Matsushita, N. and Rekimoto, J. 1997. HoloWall: designing a finger, hand, body, and object sensitive wall. In Proceedings of the 10th Annual ACM Symposium on User interface Software and Technology (Banff, Alberta, Canada, October 14 - 17, 1997). UIST '97. ACM, New York, NY, pp. 209-210.
- [Maury 99a] Maury, S., Athènes, S. & Chatty, S. 1999. Vers une interaction basée sur le rythme : les menus rythmiques. Publié dans les actes d'IHM99.
- [Maury 99b] Maury, S., Athènes, S., and Chatty, S. 1999. Rhythmic menus: toward interaction based on rhythm. In CHI '99 Extended Abstracts on Human Factors in Computing Systems (Pittsburgh, Pennsylvania, May 15 - 20, 1999). CHI '99. ACM, New York, NY, 254-255.
- [Maya] <http://usa.autodesk.com>
- [MicrosoftSurf] <http://www.microsoft.com/SURFACE/index.html>
- [McCall 77] McCall, J. 1977. Factors in Software Quality. General Electric Eds.
- [Miller 5] Miller, G. A. 1956. The magical number seven plus or minus two: Some limits on our capacity for processing information. Psychological Review, 63, pp. 81-97.
- [Moyle 02] Moyle, M. and Cockburn, A. 2002. Analysing Mouse and Pen Flick Gestures. In Proceedings of the SGCHI-NZ Symposium On Computer-Human Interaction.
- [Myers 98] Myers, B. A. 1998. A brief history of human-computer interaction technology. interactions 5, 2 (Mar. 1998), pp. 44-54.
- [Nancel 08] Nancel, M., Beaudouin-Lafon, M. 2008. Extending Marking menus with intégral dimensions : application to the dartboard menu. Rapport de recherche du LRI, n°1503.
- [Nigay 98] Nigay L, Vernier F. 1998. Navigational interaction techniques in the search results space. Paper presented at the Workshop on Innovation and Evaluation in Information Exploration Interfaces, Berkeley, CA, 1998.
- [Nielsen 90] Nielsen, J., Molich, R. 1990. Heuristic évaluation of user interfaces. Proceedings of the CHI'90 Conference on Computer Human Interaction, Seattle, ACM New York, pp. 349-356.

- [Nielsen 93] Nielsen, J. 1993. Usability Engineering, Academic Press Professional, 362 pages.
- [Norman 88] Norman, K. L., 1988. User response time for pulldown menus as a function of knowledge of location and structure of menu. Working paper, Automation Psychology Lab, University of Maryland, College Park, MD.
- [Norman 91] Norman, K. L. 1991. The Psychology of Menu Selection: Designing Cognitive Control at the Human/Computer Interface. Greenwood Publishing Group Inc.
- [Norman 99] Norman, D. A. 1999. Affordance, conventions, and design. interactions 6, 3 (May. 1999), pp.38-43.
- [Norman 04] Norman, D. A. (2004). Emotional design: Why we love (or hate) everyday things. New York: Basic Books.
- [Oakley 07] Oakley, I. and Park, J. 2007. A motion-based marking menu system. In CHI '07 Extended Abstracts on Human Factors in Computing Systems (San Jose, CA, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, pp. 2597-2602.
- [Odell 04] Odell, D. L., Davis, R. C., Smith, A., and Wright, P. K. 2004. Toolglasses, marking menus, and hotkeys: a comparison of one and two-handed command selection techniques. In Proceedings of Graphics interface 2004 (London, Ontario, Canada, May 17 - 19, 2004). ACM International Conference Proceeding Series, vol. 62. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, pp. 17-24.
- [Hodges 07] Hodges, S., Izadi, S., Butler, A., Rrustemi, A., and Buxton, B. 2007. ThinSight: versatile multi-touch sensing for thin form-factor displays. In Proceedings of the 20th Annual ACM Symposium on User interface Software and Technology (Newport, Rhode Island, USA, October 07 - 10, 2007). UIST '07. ACM, New York, NY, pp. 259-268.
- [Optimus] <http://www.artlebedev.com/>
- [Ouhaddi 98] Ouhaddi, H. et Horain, P. 1998. Conception et ajustement d'un modèle 3D articulé de la main. Actes des 6èmes journées du Groupe de Travail Réalité Virtuelle, Issy-les-Moulineaux, France, 12-13 mars 1998, p. 83-90.
- [Parhi 06] Parhi, P., Karlson, A. K., and Bederson, B. B. 2006. Target size study for one-handed thumb use on small touchscreen devices. In Proceedings of the 8th Conference on Human-Computer interaction with Mobile Devices and Services (Helsinki, Finland, September 12 - 15, 2006). MobileHCI '06, vol. 159. ACM, New York, NY, 203-210.

- [Pascoe 00] Pascoe, J., Ryan, N., and Morse, D. 2000. Using while moving: HCI issues in fieldwork environments. *ACM Trans. Comput.-Hum. Interact.* 7, 3 (Sep. 2000), 417-437.
- [Peres 04] Peres, S. C., Tamborello, F. P., Fleetwood, M. D., Chung, P., Paige-Smith, D. L. 2004. Keyboard Shortcut Usage : The Roles of Social Factors and Computer Experience. *Journal of Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 48 (5), pp. 803-807.
- [Plaisant 02] Plaisant, C., Grosjean, J., and Bederson, B. B. 2002. SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. In *Proceedings of the IEEE Symposium on information Visualization (infovis'02)* (October 28 - 29, 2002).
- [Pook 00] Pook, S., Lecolinet, E., Vaysseix, G., and Barillot, E. 2000. Control menus: execution and control in a single interactor. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems (The Hague, The Netherlands, April 01 - 06, 2000)*. *CHI '00*. ACM, New York, NY, pp. 263-264.
- [Potter 88] Potter, R. L., Weldon, L. J., and Shneiderman, B. 1988. Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Washington, D.C., United States, May 15 - 19, 1988)*. J. J. O'Hare, Ed. *CHI '88*. ACM, New York, NY, 27-32.
- [Raskin 00] Raskin, J. 2000 *The Humane Interface: New Directions for Designing Interactive Systems*. *ACM Press/Addison-Wesley Publishing Co.*
- [Rasmussen 83] Rasmussen, J. 1983. Skills, Rules and Knowledge: Signals, Signs and Symbols and other Distinctions in Human Performance Models. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13, 257-266.
- [Rekimoto 03] Rekimoto, J., Ishizawa, T., Schwesig, C., and Oba, H. 2003. PreSense: interaction techniques for finger sensing input devices. In *Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology (Vancouver, Canada, November 02 - 05, 2003)*. *UIST '03*. ACM, New York, NY, 203-212.
- [Ren 08] Ren, X., Yin, J., Oya, T., and Liu, Y. 2008. Enhancing Pie-Menu Selection with Pen Pressure. In *Proceedings of the 2008 3rd international Conference on innovative Computing information and Control - Volume 00 (June 18 - 20, 2008)*. *ICICIC*. IEEE Computer Society, Washington, DC, 364.
- [Roudaut 08] Roudaut, A., Huot, S., and Lecolinet, E. 2008. TapTap and MagStick: improving one-handed target acquisition on small touch-screens. In *Proceedings of the Working Conference on Advanced Visual*

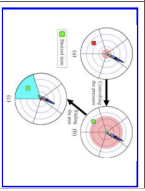
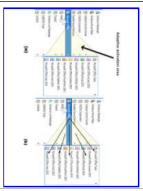
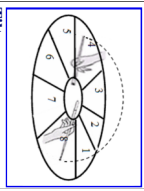
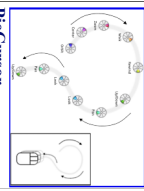
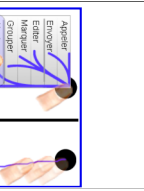
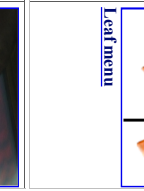
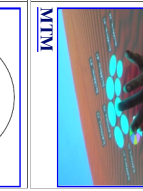
interfaces (Napoli, Italy, May 28 - 30, 2008). AVI '08. ACM, New York, NY, pp. 146-153.

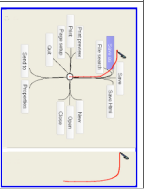
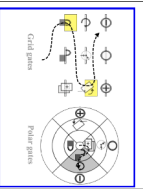



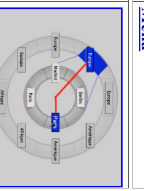
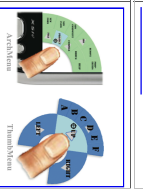

- [Roudaut 09] Roudaut A., Lecolinet E., Guiard and Y. 2009. MicroRolls: Expanding Touch-Screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb . Conference Proceedings of CHI'09, ACM Press.
- [Rubio 02] Rubio, J.M. and Janecek, P. 2002. Floating Pie Menus : Enhancing the functionality of Contextual Tools. Poster UIST'02.
- [Scapin 86] Scapin, D. L. 1986. Guide ergonomique de conception des interfaces homme-machine. Rapport technique de l'INRIA-Rocquencourt. 92 pages.
- [Schumacher 95] Schumacher, R. M., Hardzinski, M.L., Schwartz, A.L. 1995. Increasing the Usability of Interactive Voice Response Systems: Research and Guidelines for Phone-Based Interfaces. Human Factors: The Journal of the Human Factors and Ergonomics Society 37 (June), pp.251-264.
- [Sears 94] Sears, A. and Shneiderman, B. 1994. Split menus: effectively using selection frequency to organize menus. ACM Trans. Comput.-Hum. Interact. 1, 1 (Mar. 1994), pp. 27-51.
- [secondlife] <https://jira.secondlife.com/browse/VWR-5067>
- [Sellen 90] Sellen, A., Kurtenbach, G., and Buxton, W. 1990. The role of visual and kinesthetic feedback in the prevention of mode errors. In Proceedings of the IFIP Tc13 Third interational Conference on Human-Computer interaction (August 27 - 31, 1990). D. Diaper, D. J. Gilmore, G. Cockton, and B. Shackel, Eds. North-Holland Publishing Co., Amsterdam, The Netherlands, pp. 667-673.
- [Senach 90] Senach, B. 1990 Evaluation ergonomique des interfaces homme-machine : une revue de la littérature. Rapport de recherche INRIA n°1180, Programme 8, Communication Homme-Machine.
- [Shneiderman 92] Shneiderman, B. 1992. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison Wesley; 2th edition.
- [Shneiderman 04] Shneiderman, B. and Plaisant, C. 2004. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison Wesley; 4th edition.
- [Simon 76] Simon, H. 1976. Administrative behavior: a study of décision making processes in administrative organization (3rd ed.) New York: Free Press.
- [Smith 86] Smith, S.,L., Mosier, J.N. 1986. A design évaluation checklist for user-system interface software, Report #MTR-9480 EDS_TR_84-358, The MITRE Corporation, Bedford, MA, 1982.
- [Snapshots] <http://www.snap.com/snapshots.php>

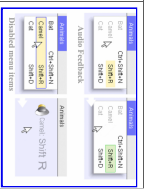
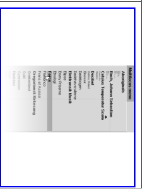



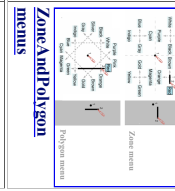


- [Snowberry 83] Snowberry, K., Parkinson, S. R., and Sisson, N. 1983. Computer display menus. *Ergonomics*, 26, pp. 699-712.
- [Snyder 85] Snyder, K. M., Happ, A. J., Malcus, L., Paap, K. R. and Lewis, J. R. 1985. Using cognitive models to create menus. *Proceedings of the Human Factors Society 29th Annual Meeting*. Santa Monica, CA : Human Factors Society. pp. 655-658.
- [Soliz 03] Soliz, E., Paley, W.B. 2003. A Re-Interpretation of marking Menus: The Usage of Gestalt Theory as Cognitive Tools. Poster of the 16th Annual ACM Symposium on User Interface and Software Technology. ACM UIST'03.
- [Tanvir 08] Tanvir, E., Cullen, J., Irani, P., and Cockburn, A. 2008. AAMU: adaptive activation area menus for improving selection in cascading pull-down menus. In *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems (Florence, Italy, April 05 - 10, 2008)*. CHI '08. ACM, New York, NY, pp. 1381-1384.
- [Tapia 95] Tapia, M. A. and Kurtenbach, G. 1995. Some design refinements and principles on the appearance and behavior of marking menus. In *Proceedings of the 8th Annual ACM Symposium on User interface and Software Technology (Pittsburgh, Pennsylvania, United States, November 15 - 17, 1995)*. UIST '95. ACM, New York, NY, pp. 189-195.
- [Taylor 76] Taylor, D. A. 1976. Holistic and analytic processes in the comparison of letters. *Perception & psychophysics*, 20, pp. 187-190.
- [Tian 08] Tian, F., Xu, L., Wang, H., Zhang, X., Liu, Y., Setlur, V., and Dai, G. 2008. Tilt menu: using the 3D orientation information of pen devices to extend the selection capability of pen-based user interfaces. In *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems (Florence, Italy, April 05 - 10, 2008)*. CHI '08. ACM, New York, NY, pp. 1371-1380.
- [TackpadGestures] <http://support.apple.com/kb/HT1115>
- [Tsandilas 07] Tsandilas, T. and schraefel, m. c. 2007. Bubbling menus: a selective mechanism for accessing hierarchical drop-down menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007)*. CHI '07. ACM, New York, NY, pp. 1195-1204.
- [TutorielMaya] <http://www.cayus.net/maya/maya-tips-du-jour-maximiser-lespace-de-travail/>
- [VistaStartMenu] <http://www.prweb.com/releases/ordinarysoft/VistaStart/>


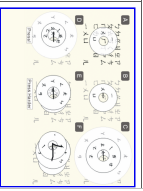
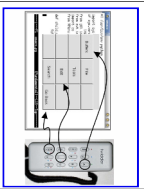
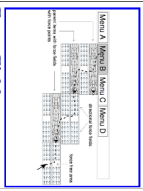
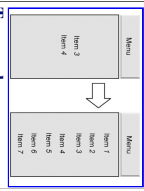

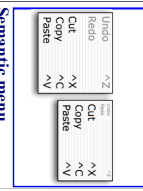
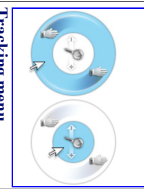
- [Viviani 82] Viviani, P., Terzuolo, C. 1982. Trajectory determines movement dynamics. In *Neuroscience*, 7(2), pp. 17-24.
- [Vogel 07] Vogel, D. and Baudisch, P. 2007. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007)*. CHI '07. ACM, New York, NY, 657-666.
- [Walker 90] Walker, N. and Smelcer, J. B. 1990. A comparison of selection time from walking and pull-down menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Empowering People (Seattle, Washington, United States, April 01 - 05, 1990)*. J. C. Chew and J. Whiteside, Eds. CHI '90. ACM, New York, NY, 221-226.
- [Waterworth 91] Waterworth, J., Chignell, M. 1991. Model of Information Exploration. *Hypermedia*, 3(1), 1991, pp. 35-38.
- [Welie 99a] Welie, M.V., Van der Veer, G.C., Elins A. 1999. Usability Properties in Dialog Models. In *6th International Eurographics Workshop on Design Specification and Verification of Interactive Systems DSV-IS99*, pp. 2-4.
- [Welie 99b] Welie, M.V., Van der Veer, G.G., Eliens, A. 1999. Breaking down usability. In *proceeding of Interact'99*, pp. 613-620.
- [Wellner 93] Wellner, P. 1993. Interacting with paper on the DigitalDesk. *Commun. ACM* 36, 7 (Jul. 1993), pp. 87-96.
- [Widgor 07] Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. 2007. Lucid touch: a see-through mobile device. In *Proceedings of the 20th Annual ACM Symposium on User interface Software and Technology (Newport, Rhode Island, USA, October 07 - 10, 2007)*. UIST '07. ACM, New York, NY, 269-278.
- [Wiseman 69] Wiseman, N. E., Lemke, H. U., Hiles, J. O. 1969 *PIXIE: A New Approach to Graphical Man-machine Communication*. CAD Conf. Southampton 463 IEEE Conference Publication 51, pp. 463.
- [Witten 84] Witten, I.H., Cleary, J.G., Greenberg, S. 1984. On Frequency-Based Menu-Splitting algorithms. *International Journal of Man-Machine Studies* 21(2), pp. 135-148.
- [Wu 03] Wu, M. and Balakrishnan, R. 2003. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology (Vancouver, Canada, November 02 - 05, 2003)*. UIST '03. ACM, New York, NY, 193-202.
- [Yates 66] Frances Yates (trad. Daniel Arasse), *L'Art de la Mémoire (The Art of Memory)*, éditions Gallimard, coll. « nrf », Paris, 1966 (réimpr. 1987)


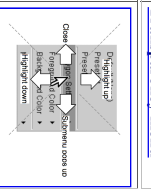
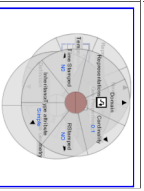
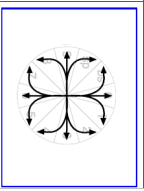

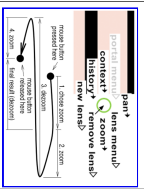
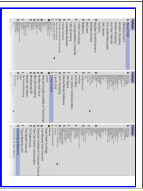
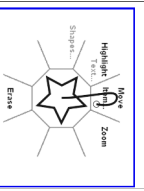

- [Yin 06] Yin, M. and Zhai, S. 2006. The benefits of augmenting telephone voice menu navigation with visual browsing and search. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI '06. ACM, New York, NY, pp. 319-328.
- [Zaphiris 02] Zaphiris, P. 2002 Age Differences and the Depth-Breadth Tradeoff in Hierarchical Online Information Systems. Doctoral Thesis. UMI Order Number: AAI3047597., Wayne State University.
- [Zhao 04] Zhao, S. and Balakrishnan, R. 2004. Simple vs. compound mark hierarchical marking menus. In Proceedings of the 17th Annual ACM Symposium on User interface Software and Technology. ACM UIST '04, pp. 33-42.
- [Zhao 06] Zhao, S., Agrawala, M., and Hinckley, K. 2006. Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI '06. ACM, New York, NY, pp. 1077-1086.
- [Zhao 07] Zhao, S., Dragicevic, P., Chignell, M., Balakrishnan, R., and Baudisch, P. 2007. Earpod: eyes-free menu selection using touch input and reactive audio feedback. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 1395-1404.
- [Zhao 08] Zhao, S., 2008. Earpod: Efficient Hierarchical Eyes-free Menu selection. University of Toronto, 2008, Ph. D., Thesis, Dept. Of Computer Science.
- [Zhao 07b] Zhao, S., Chignell, M. 2008. Using Glyphs To Facilitate Transition From Hierarchical Selection To Gesturing. 10th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems (IFAC-HMS 2007), Seoul, Korea.
- [Zimmerman 87] Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., and Harvill, Y. 1987. A hand gesture interface device. SIGCHI Bull. 18, 4 (Apr. 1987), 189-192.
- [Zipf 49] Zipf, G.K. 1949. Human Behaviour and the Principles of Least Effort. Ontario : Addison-Wesley.

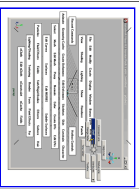
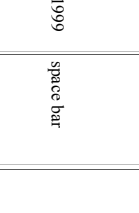
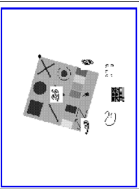
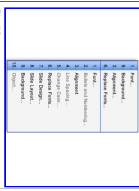
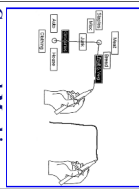
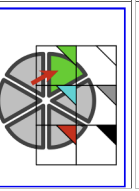
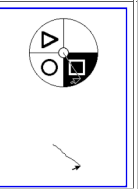
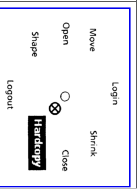
	2008			circular design		stroke shortcuts												24 items (orientation+pressure)	4-5 menu levels		pressure input et interactive surface		eyes-free selection	
	2008	previsualization	enlarged submenu activation area		keyboard shortcuts	keyboard shortcuts	previsualization	keyboard shortcuts										~30 items	1 menu level		mouse et pen			merging (transition command value)
	2008		circular design	crossing (on the altitude)	no expert mode	no expert mode		no expert mode										8 items	1 menu level		pen with 3D orientation			
	2008	click through paradigm	circular design		no expert mode	no expert mode		no expert mode										8 items	1 menu level		mouse et pen			merging (transition command value)
	2008	previsualization	larger item (for finger tap) et mirror effect (for frequent items)		stroke shortcuts	stroke shortcuts		stroke shortcuts										7 items	3 menu levels	total	handheld devices, mobile devices, interactive surfaces, et redundancy mode (for expert mode selection in close proximity of screen borders)	handheld devices, mobile devices, interactive surfaces, et mobile devices, et interactive surfaces	eyes-free selection	
	2008	the heel of the hand	finger affection		hand configuration	hand configuration	previsualization	hand configuration										8 items	2 menu levels		multi-touch surfaces	multi-touch surfaces		rapid combination of commands
	2008		circular design		stroke shortcuts	stroke shortcuts		stroke shortcuts										24 items (orientation+distance)			mouse et pen		eyes-free selection	

	2008		Within group	circular design		stroke shortcuts		within group	stroke shortcuts	gestalt principles				20 items et curved gestures	3, 4 menu levels, et temporal composition of marks		pen et interactive surface	interactive surface	eyes-free selection	superimposed marks (transition command object)
	2008			circular design et grid design	crossing	no expert mode			no expert mode				9-12 items	1 menu level		pen et interactive surface	interactive surface		merging (transition command value) et modal (transition command command)	
	2007		highlighted frequent items	bubble cursor (frequent items)		keyboard shortcuts			keyboard shortcuts				~30 items	4-5 menu levels		mouse et pen				
	2007			circular design		stroke shortcuts			stroke shortcuts				8 items	2-3 menu levels		touchpad	audio			
	2007		occlusion avoided et custom path			no expert mode			no expert mode				~30 items	1 menu level		interactive surfaces et physical objects	interactie surfaces			
	2007		previsualization et simple navigation	circular design		stroke shortcuts		previsualization et simple navigation	stroke shortcuts	gestalt principle			8 et 12 items	3, 4 menu levels, et temporal composition of marks				concentric layout et eyes-free selection	superimposed marks	
	2007		previsualization et occlusion avoided	circular design et offset cursor		no expert mode			no expert mode				4-5 items	2 menu levels		interactive surfaces (thumb) et mobile devices	interactive surfaces			
	2007		adaptive item placement	adaptive item placement		keyboard shortcuts			keyboard shortcuts				30 items	4-5 menu levels		mouse et pen				

	2007		previsualization		keyboard shortcuts		previsualization	keyboard shortcuts									~30 items	4-5 menu levels	mouse et pen					
	2007		highlighted frequent items		keyboard shortcuts		keyboard shortcuts	keyboard shortcuts									~50 items (fisheye view)	4-5 menu levels	mouse et pen					
	2007		enlarged frequent items	enlarged frequent items	keyboard shortcuts		keyboard shortcuts	keyboard shortcuts									~30 items	4-5 menu levels	mouse et pen					
	2007	both index and thumb finger are detected on the surface		finger span	finger configuration		finger configuration	finger configuration									4-16 items	1 menu level	multi-touch surfaces	multi-touch surfaces	eyes-free selection		merging (transition command value) et modal (transition command command)	
	2006		circular design	circular design	stroke shortcuts		stroke shortcuts	stroke shortcuts									8 items	2-3 menu levels	mouse et pen		eyes-free selection		transition command object	
	2006		circular design et scale independence		stroke shortcuts		stroke shortcuts	stroke shortcuts									16/32 items (orientation+relative position)	3-4 menu levels (temporal composition of marks)	mouse et pen		eyes-free selection			
	2006	two fingers	circular design	crossing	no expert mode		no expert mode	no expert mode									6 items	1 menu level	interactive surfaces (finger)	interactive surfaces				
	2006		jumping in submenu		keyboard shortcuts		keyboard shortcuts	keyboard shortcuts									~30 items	4-5 menu levels	mouse et pen					

	2006	button (in the non-dominant hand)	semi-circular design	no expert mode	no expert mode	9 items	2 menu levels	tablet pc et pen	tablet pc	command-command
	2005		circular design			24 items (orientation+pressure)		pressure input et interactive surface		
	2005		finger mapping	tap shortcuts	tap shortcuts	12 items	2-3 menu levels	Twiddler one-handed wearable keyboard		eyes-free selection
	2005	previsualization et within groups	force field	keyboard shortcuts	keyboard shortcuts	~30 items	4-5 menu levels	mouse		
	2004	temporal priority, spatial consistency, et previsualization		keyboard shortcuts	keyboard shortcuts	30 items	3-4 menu levels	mouse et pen		
	2004		circular design	stroke shortcuts	stroke shortcuts	8 items	3-4 menu levels (temporal composition of marks)	mouse et pen	output	eyes free selection et superimposed menu levels
	2004		semantic pointing	keyboard shortcuts	keyboard shortcuts	~30 items	4-5 menu levels	mouse et pen		
	2003		circular design	no expert mode	no expert mode	8 items	1 menu level	pen et tablet pc	table pc	modal (transition command) et merging (transition command value)

	2003		previsualization of increasing the transparency of infrequently items		keyboard shortcuts		previsualization	keyboard shortcuts												~30 items	4-5 menu levels	mouse et pen						
	2003		submenu is located at the position where the horizontal motion occurred		keyboard shortcuts			keyboard shortcuts												~30 items	4-5 menu levels	mouse et input						
	2002				no expert mode			no expert mode												8 items	2-3 menu levels	mouse et pen						modal (transition command-command)
	2002				stroke shortcuts			stroke shortcuts												12 items	2-3 menu levels	touchpad et fingers						eyes-free selection
	2002		both hands on the desk and all fingers of the left hand closed		no expert mode			no expert mode												~5 items	2 menu levels	interactive surfaces						
	2000				stroke shortcuts			stroke shortcuts												8 items	2-3 menu levels	mouse et pen						merging command selection and direct manipulation (transition command-value)
	2000				keyboard shortcuts			keyboard shortcuts												~50 items	4-5 menu levels	mouse et pen						
	2000				stroke shortcuts			stroke shortcuts												8 items	2-3 menu levels	mouse et pen						merging command selection and direct manipulation (transition command-value)
	2000				stroke shortcuts			stroke shortcuts												8 items	2-3 menu levels	mouse et pen						merging command selection and direct manipulation (transition command-value)

	1999			rythm	no expert mode													~8items	1 menu level	mouse et pen				
	1999	space bar		linear and circular design	keyboard shortcuts et stroke shortcuts				keyboard shortcuts et stroke shortcuts											mouse, pen, et keyboard				
	1997	click through paradigm			no expert mode				no expert mode											two handed interaction (mouse + trackball et ...)				merging (transition command value)
	1994		frequent items are copied to an adaptive section (top of the menu)	frequent items are copied to an adaptive section (top of the menu)	keyboard shortcuts				keyboard shortcuts									~30 items	3-4 menu levels	mouse et pen				
	1993			circular design et scale independence	stroke shortcuts				stroke shortcuts									8 items	2 menu levels	mouse et pen				
	1993	click through paradigm			no expert mode				no expert mode									8 items	1 menu level	two handed interaction (mouse + trackball et ...)				merging (transition command value)
	1991			circular design et scale independence	stroke shortcuts				stroke shortcuts									8 items	1 menu level	mouse et pen				eyes-free selection
	1988			circular design	no expert mode				no expert mode									8 items	2-3 menu levels	mouse et pen				

Résumé

Les menus permettent d'explorer et de sélectionner les commandes d'une application. Ils sont présents dans la plupart des applications et concernent un large public. Les menus sont donc au cœur de l'Interaction Homme-Machine (IHM). Aussi, cet enjeu a motivé de nombreux travaux en IHM : face au foisonnement de nouvelles techniques de menus, il n'est pas facile d'appréhender les avancées et de cerner les différences et similitudes entre les techniques de menus conçues. Dans ce contexte, cette thèse en IHM propose un espace de conception des techniques de menus, appelé **MenUA**. Cet espace repose sur une liste de critères utiles pour structurer les travaux sur les menus et permettre aux concepteurs d'applications de faire des choix informés. MenUA facilite aussi l'exploration de l'espace des possibilités lors de la conception : il a été à l'origine de quatre nouvelles techniques de menus que nous avons conçues, développées et évaluées : les Wave menus, les Flower menus, les Leaf menus et les Multi-Touch Menus. Les **Wave menus** améliorent le mode novice des Marking menus en favorisant la navigation dans la hiérarchie de commandes. En considérant la courbure des gestes de l'utilisateur, les **Flower menus** augmentent le nombre de commandes disponibles et améliorent l'apprentissage du mode expert des Marking menus. Les **Leaf Menus** sont quant à eux des menus linéaires enrichis de raccourcis gestuels et conçus pour être utilisés sur des dispositifs mobiles. Enfin, les **Multi-Touch Menus** exploitent la technologie multi-points des surfaces tactiles afin de naviguer et sélectionner des commandes avec les cinq doigts de la main.

Mots-clés

Interaction Homme-Machine, IHM, Techniques d'interaction, Techniques de menus, systèmes de menus.

Abstract

Menus are used for exploring and selecting commands in interactive applications. They are widespread in current applications and used by a large variety of users. As a consequence, menus are at the heart of Human-Computer Interaction (HCI) and motivate many studies in HCI. Facing the large variety of designed menu techniques, it is however difficult to have a clear understanding of the design possibilities, to understand the advances as well as to compare existing menu techniques. In this context, this thesis in HCI proposes a design space of menu techniques called **MenUA**. MenUA is based on a list of criteria that define a coherent framework of design issues for menus. MenUA also helps application designers to make informed design choices and is a support for exploring design alternatives. Stemming from MenUA, we designed, developed and evaluated four menu techniques: Wave menus, Flower menus, Leaf menus and Multi-Touch Menus. **Wave menus** improve the novice mode of Marking menus by making the navigation within the hierarchy of commands easier. **Flower menus** increase the menu breadth of Marking menus while supporting good learning performance of the expert mode. **Leaf menus** are linear menus enriched by stroke shortcuts to facilitate the selection of commands on small handheld touch-screen devices. Finally, **Multi-Touch Menus** exploit the recent capabilities of multi-touch surfaces in order to allow the users to explore and select commands using the five fingers of the hands.

Keywords

Human Computer Interaction, HCI, Interaction techniques, menu techniques, menu systems.