



Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in:
<http://oatao.univ-toulouse.fr/19175>

Official URL: <https://dl.acm.org/citation.cfm?doid=3132129.3132131>

DOI : <http://doi.org/10.1145/3132129.3132131>

To cite this version: Dubois, Emmanuel and Raynal, Mathieu and Serrano, Marcos *Améliorer l'interaction avec une barre de menus grâce à des gestes d'inclinaison sur une souris multidimensionnelle*. (2017) In: 29eme Conference francophone sur l'Interaction Homme-Machine (IHM 2017), 29 August 2017 - 1 September 2017 (Poitiers, France).

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Améliorer l'interaction avec une barre de menus grâce à des gestes d'inclinaison sur une souris multidimensionnelle

Taking advantage of rolling gestures on a multidimensional mouse for interacting with a menubar

Emmanuel Dubois
IRIT, Université de Toulouse
31062, Toulouse
Cedex, France
Emmanuel.Dubois@irit.fr

Mathieu Raynal
IRIT, Université de Toulouse
31062, Toulouse
Cedex, France
Mathieu.Raynal@irit.fr

Marcos Serrano
IRIT, Université de Toulouse
31062, Toulouse
Cedex, France
Marcos.Serrano@irit.fr

ABSTRACT

This paper presents Rolling-Menu, an interaction technique based on the use of a multidimensional device, the Roly-Poly Mouse (RPM), to interact with a menu bar. Rolling-Menu reduces the distance covered with the pointer between the point of interaction in the application and the menu bar: this results in a better integration between the command selection and the direct manipulation of the content of an application. In addition, selecting an item from a menu bar with Rolling-Menu requires to roll RPM in a predefined direction. Therefore, such physical manipulations ensure that a similar access time is required to activate every menu item. An experiment evaluates different implementations of Rolling-Menu and compares them with a mouse. The results establish that the validation mode and the mapping between the roll direction and the items influence the performance of Rolling-Menu. It also shows that the best version of Rolling-Menu is between 14% and 40% faster than the mouse to select an item in a menubar containing 4 to 10 items. Finally, the error rate with Rolling-Menu (4%) is similar to that obtained with the mouse.

KEYWORDS

Menu bar, Multidimensional device, Roly-Poly Mouse, hemispherical mouse.

RÉSUMÉ

Cet article présente Rolling-Menu, une technique d'interaction basée sur l'utilisation d'un dispositif multidimensionnel, la souris hémisphérique Roly-Poly Mouse (RPM), pour interagir avec une barre de menus. Rolling-Menu contribue à réduire la distance parcourue par le pointeur entre le point d'interaction dans l'application et la barre de menus : il en résulte une meilleure intégration entre la sélection de commande et la manipulation directe du contenu d'une application. De plus, sélectionner un menu dans une barre de menus repose sur une inclinaison de la RPM dans une direction prédéfinie. De telles manipulations physiques demandent donc un temps d'accès similaire pour chaque menu. Une expérimentation évalue différentes implémentations de Rolling-Menu et les compare avec une souris. Les résultats établissent que le mode de validation et le mapping entre les inclinaisons et les items influencent la performance de Rolling-Menu. Il ressort également que la meilleure technique de Rolling-Menu est entre 14% et 40% plus rapide que la souris pour sélectionner un menu dans une barre de menus contenant de 4 à 10 menus. Enfin le taux d'erreur avec Rolling-Menu (4%) est similaire à celui obtenu avec la souris.

MOTS-CLEFS

Menu linéaire, dispositif multidimensionnel, Roly-Poly Mouse, souris hémisphérique.

1 INTRODUCTION

Les barres de menus, ou menus linéaires [10], sont quasi omniprésentes en environnement de bureau. Elles sont composées d'un ensemble de menus sélectionnables : sélectionner un menu ouvre alors une liste contenant des sous-items. L'utilisation d'une barre de menus repose majoritairement sur l'utilisation d'une souris et de son pointeur : elle nécessite alors une tâche de pointage (pour atteindre un menu ou un sous-item) suivie d'une tâche de validation (pour sélectionner le menu ou le sous-item). Par conséquent, la sélection d'un menu induit une rupture dans le flux de l'interaction : en effet, un utilisateur qui réalise une tâche dans une application doit déplacer le curseur vers le haut de l'écran

<https://doi.org/10.1145/3132129.3132131>

pour sélectionner un des menus de la barre de menus, puis ramener le curseur à sa position initiale pour poursuivre sa tâche.

Pour limiter cette rupture entre sélection de menu et utilisation d'une application, de multiples solutions ont été étudiées. Les raccourcis clavier sont la solution la plus évidente à ce problème mais nécessitent d'être mémorisés, et ne sont donc souvent utilisés que pour les commandes les plus fréquentes. Plusieurs techniques d'interaction optimisent la sélection d'un item [8,11,23], mais elles nécessitent toujours des déplacements du curseur entre application et menu. Les menus contextuels circulaires [4,10,17,24] ou linéaires [19] limitent efficacement ce déplacement, mais ils ne sont pas visibles avant leur activation et ils ne proposent qu'un sous-ensemble des menus contenus dans la barre de menus globale, en fonction du lieu d'activation.

Pour mieux tirer profit de la visibilité et de la stabilité du contenu des barres de menus, il faudrait donc limiter la distance parcourue par le pointeur entre le point d'interaction dans l'application et la barre de menus : une telle optimisation contribuera à offrir une meilleure intégration entre la sélection de commande et la manipulation directe du contenu d'une application.

Dans cet article nous explorons l'intérêt des souris multidimensionnelles pour optimiser l'utilisation d'une barre de menus horizontale : offrant plus de degrés de libertés (DDL) qu'une simple souris, certains DDL d'une souris multidimensionnelle peuvent être dédiés au contrôle du menu, tandis que d'autres seront réservés à l'interaction avec l'application [31]. De nombreuses souris à plusieurs DDL ont été proposées pour interagir avec des environnements 3D ou multidimensionnels, comme par exemple la Rockin'Mouse [6], la VideoMouse [20], la Roly-Poly Mouse [26,27] ou le Z mouse [36] présenté au CES 2017. Néanmoins, aucun de ces projets ne s'est intéressé à l'intérêt de tels dispositifs pour interagir avec les menus linéaires.

Nos travaux ont pour objectif d'étudier les apports d'une souris multidimensionnelle, la Roly-Poly Mouse [26,27] (RPM), susceptible de remplacer la souris classique, pour interagir avec une barre de menus en environnement de bureau. Nous nous intéressons en particulier à la possibilité d'activer directement les menus d'une barre de menus au travers de l'utilisation des inclinaisons de cette souris multidimensionnelle : chaque menu est alors associé à l'inclinaison de RPM dans une direction prédéfinie et ne nécessite plus de déplacer le curseur vers la barre de menus pour être sélectionné.

Afin d'étudier l'impact du mapping entre les inclinaisons et les menus d'une part, et l'impact du mode de validation d'autre part, nous avons conçu et implémenté quatre versions de Rolling-Menu, des techniques d'interaction basées sur l'inclinaison de la RPM. Au travers d'une expérimentation utilisateur, nous comparons ensuite ces quatre Rolling-Menu avec l'utilisation d'une souris classique. Les résultats mettent en évidence l'effet du mode de validation sur la performance des Rolling-Menu : l'utilisation d'un bouton s'avère plus performant que la validation par inclinaison totale. Ils établissent également que les Rolling-Menu sont toujours plus rapides que la souris traditionnelle pour sélectionner un menu contenant jusqu'à 4 ou 6 items. Au-delà, certaines

configurations de Rolling-Menu demeurent plus efficaces que la souris pour une barre de menus contenant jusqu'à 10 menus.

Nos contributions sont 1) la description de l'espace de conception des Rolling-Menu ; 2) l'implémentation de plusieurs versions de ces techniques ; 3) la validation expérimentale de ces techniques.

2 ÉTAT DE L'ART

Pour optimiser la transition objet - commande [2], i.e. minimiser l'effort à produire pour passer de la zone de travail courante à la zone d'activation d'une commande, de nombreux travaux ont porté sur la conception de menus circulaires et contextuels [1,18,29,35]. Toutefois de telles solutions n'assurent pas la visibilité permanente des commandes disponibles, n'offrent souvent qu'une sous partie des menus / commandes de l'application ou posent des problèmes d'occlusion (dans les coins, par le pouce, etc.). Dans cet article nous nous concentrons sur l'interaction avec un menu classique, stable, visible en permanence et placé en haut d'une fenêtre de travail. Dans ce contexte, l'utilisation de techniques d'interaction multidimensionnelles pour pointer et sélectionner les éléments d'un menu permet de différencier les dimensions de manipulation du menu et de l'application interactive utilisée. Nous synthétisons ici les principales techniques d'interaction multidimensionnelles envisagées dans la littérature pour contrôler un menu puis nous présentons les souris multidimensionnelles existantes.

2.1 Techniques d'interaction multidimensionnelles pour menus

Plusieurs techniques d'interaction multidimensionnelles ont été appliquées dans le passé à la manipulation de menus, utilisant entre autres des stylets ou des écrans tactiles multipoints. PushMenu [21] ajoute un degré de liberté par la pression appliquée sur un stylet pour permettre l'accès à 20 menus. Le Multi-Touch Menu [3] exploite les cinq doigts de la main pour optimiser un menu hiérarchique pouvant contenir jusqu'à 12 sous-menus pour chacun des huit menus de premier niveau. Une autre approche tactile multipoint s'appuie sur la métaphore du pianiste : plaquer un accord à quatre doigts sur une surface tactile déclenche l'activation d'un élément du menu hiérarchique [25].

Une autre façon de permettre à des modalités d'interaction classiques de contrôler des dimensions supplémentaires consiste à exploiter la dimension spatiale de leur manipulation. Ainsi en plus des translations habituellement détectées sur un écran tactile, les Microrolls gestures [28] exploitent les rotations d'un doigt sur l'écran pour activer un menu. L'utilisation de la détection d'inclinaisons pour contrôler un menu ont aussi été explorées lors d'une interaction avec un stylet [31]. Enfin, si une surface tactile offre habituellement deux dimensions (x,y), l'introduction des Hover Widgets [15] a aussi démontré qu'une interaction à proximité de la surface avec un stylet est aussi contrôlable et offre ainsi de nouveaux DDL. Une solution bimanuelle et sans stylet a également été introduite plus récemment [16].

Toutefois ces interactions multidimensionnelles reposent sur l'introduction d'un dispositif complémentaire à la souris et au clavier, qu'il s'agisse d'un stylet, d'une surface tactile ou d'un

capteur de pression. En effet ces techniques d'interaction ne couvrent pas le pointage standard dans l'application interactive et ne peuvent donc se substituer à la souris.

2.2 Souris multidimensionnelles

Les souris multidimensionnelles enrichissent les fonctionnalités habituellement couvertes par une souris, avec par des capacités d'interaction supplémentaires. Elles présentent donc cette aptitude à consacrer une partie de leurs dimensions à la gestion des menus et une autre à l'interaction avec l'application. C'est par exemple le cas de LenseMouse qui accole un écran tactile sur le sommet d'une souris [34], ou encore Inflatable Mouse qui ajoute une capacité de déformation par pression de la main à une souris classique [22]. C'est aussi le cas des souris offrant des capacités multi-touch [7].

Une autre approche consiste à concevoir des dispositifs multidimensionnels s'appuyant sur des manipulations physiques distinctes. Leurs caractéristiques sont alors proches de celles de RPM [26,27]. En effet la forme hémisphérique de RPM permet d'incliner le dispositif comme Rockin'Mouse [6] et VideoMouse [20]. À l'inverse de ces deux dispositifs, la base de RPM est une véritable hémisphère, permettant des amplitudes d'inclinaison bien supérieures et des combinaisons de manipulations physiques (cf. Table 1 dans [27] pour une comparaison plus complète entre ces dispositifs). L'intérêt de RPM a notamment été démontré pour différentes tâches d'interaction [26,27,30]. Avec de tels dispositifs, chaque manipulation physique (translation, rotation, inclinaison comme définies dans [26,27,30]) peut être associée à des commandes distinctes et ainsi offrir de multiples possibilités pour accéder à différentes commandes.

Plusieurs dispositifs multidimensionnels ont donc fait l'objet de travaux de recherche dans la littérature. Bien qu'ils présentent des caractéristiques particulièrement adaptées au contrôle optimisé d'une barre de menus, aucun d'entre eux n'a, à notre connaissance, fait l'objet d'une étude utilisateur visant à établir ses apports et limites dans ce contexte.

3 ROLLING-MENU

Nous présentons Rolling-Menu, une technique d'interaction permettant de sélectionner les menus d'une barre de menus en utilisant les inclinaisons d'une souris multidimensionnelle.

3.1 Inclinaison de RPM

Afin de sélectionner les menus d'une barre de menus, nous avons choisi d'explorer l'utilisation des inclinaisons de la Roly-Poly Mouse (RPM), une souris multidimensionnelle qui peut être translaturée, inclinée et tournée. L'utilisation des inclinaisons est motivée par le fait que les translations de la RPM servent déjà à déplacer le curseur de l'application, en remplacement de la souris, et que les rotations sont limitées en amplitude et jugées peu confortables [26]. L'inclinaison de la RPM se décline en deux valeurs qui peuvent être exploitées pour l'interaction :

- La *direction* d'inclinaison : elle indique la direction, vue d'en haut, dans laquelle la RPM est inclinée (Figure 1 - gauche). Nous utilisons des degrés pour décrire cette direction de manière précise et avons établi l'angle 0° pour la direction Nord, -90° pour l'Ouest et 90° pour l'Est.

- L'*amplitude* de l'inclinaison : elle correspond à l'angle de RPM après l'inclinaison, par rapport à sa position en équilibre à l'horizontale (Figure 1 - centre). Cet angle est aussi exprimé en degré. Lorsque la RPM est à sa position d'équilibre, cet angle est de 0°, et il vaut 90° lorsque la base supérieure de la RPM est perpendiculaire au plan sur lequel la RPM est posée.

3.2 Principe général des Rolling-Menus

Le fonctionnement général d'un Rolling-Menu s'apparente au fonctionnement d'un menu contextuel. Lors de l'activation du mode menu, le curseur de l'application se fige et la barre de menus est associée à une *plage* d'inclinaisons (Figure 1-droite). La plage est divisée en secteurs, de sorte que chaque menu de la barre de menus est associé à un *secteur* d'inclinaison de RPM (Figure 1-droite). Afin de sélectionner un menu, il suffit de réaliser une inclinaison dans la direction de son secteur. Rolling-Menu s'appuie sur un référentiel fixe dans le monde physique pour associer une direction d'inclinaison à un menu.

Une fois qu'un menu est sélectionné, le sous-menu s'ouvre et les différents sous-items peuvent être parcourus. Ce parcours dans les sous-items peut être réalisé avec RPM de la même manière qu'avec une souris, i.e. par simple translation dans la liste des sous-menus proposés, ou par des moyens tirant profit des degrés de liberté du dispositif, comme évoqué en perspectives. Dans le présent article, nous focalisons sur la sélection d'un menu de la barre de menus.

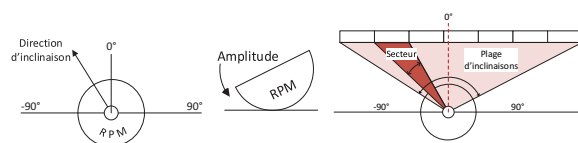


Figure 1: Gauche : direction d'inclinaison de RPM. Centre : amplitude d'inclinaison de RPM. Droite : plage et secteur d'inclinaison d'un Rolling-Menu.

3.3 Facteurs de conception

À partir de ce principe général d'interaction, nous avons conçu plusieurs versions de Rolling-Menu en variant 3 facteurs de conception : 1) la plage d'inclinaisons, 2) la taille des secteurs d'inclinaison associés à un menu, et 3) la méthode de validation.

3.3.1 *Plage d'inclinaisons*. La plage d'inclinaisons correspond à l'ensemble des directions d'inclinaisons de RPM exploitées pour interagir avec la barre de menus (Figure 1 - droite). Deux solutions sont possibles dans le cas d'une barre de menus horizontale placée en haut de la fenêtre active :

- *Plage directe* : La plage est constituée d'un triangle dont les sommets sont les deux extrémités de la barre de menus et le pointeur de la souris au moment de l'activation du Rolling-Menu (Figure 2-gauche). Ce mode offre à l'utilisateur un mapping direct entre l'inclinaison produite et la direction du menu à sélectionner par rapport au point d'activation du mode menu.
- *Plage de 180°* : La plage couvre toujours les directions de -90° à +90° (Figure 2-droite). L'utilisateur dispose ainsi d'une plage maximale pour sélectionner les items, en particulier ceux situés sur les côtés.

Nous avons écarté les plages supérieures à 180°, car nos prétests ont montré qu'il apparaît peu naturel de réaliser un geste d'inclinaison vers l'arrière pour sélectionner un item qui se trouve sur la partie supérieure de l'écran.

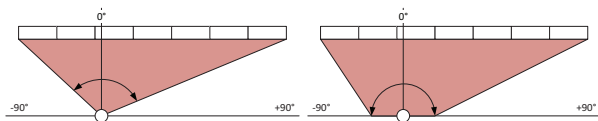


Figure 2: Plage directe (gauche) vs. plage de 180° (droite).

3.3.2 Taille des secteurs d'inclinaison. Le menu à sélectionner est désigné par le secteur d'inclinaison de RPM, comme expliqué précédemment. Trois cas se présentent selon la méthode utilisée pour diviser la plage d'inclinaisons en secteurs : les secteurs peuvent être de taille variable, de taille fixe sur toute la plage ou de taille fixe de part et d'autre de l'axe 0° de la plage (Figure 3).

a) Secteurs de taille variable : la taille des différents secteurs n'est pas la même pour chaque menu de la barre de menus et varie selon la position de déclenchement du menu (Figure 3 – ligne 1). Cette solution offre une cohérence entre l'espace moteur (l'angle à produire avec RPM) et l'espace de perception (angle entre le point d'activation du menu et les bords du menu) de l'utilisateur.

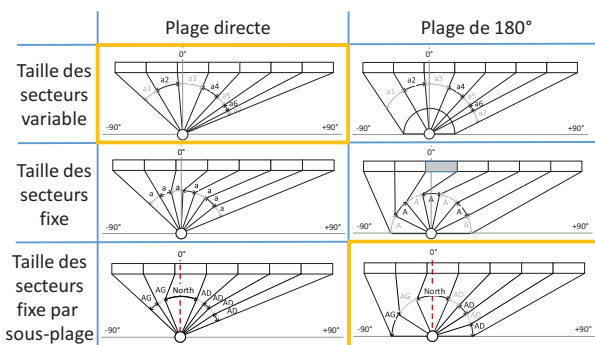


Figure 3: Secteurs de taille variable (ligne 1), fixe (ligne 2) ou fixe par sous-plage (ligne 3) pour une plage d'inclinaisons directe (gauche) et une plage de 180° (droite).

b) Secteurs de taille fixe sur toute la plage : la plage des inclinaisons dédiée au contrôle du menu est découpée en autant de secteurs que de menus potentiellement sélectionnables dans la barre de menus (Figure 3 – ligne 2). Cette solution apporte une homogénéité à la technique d'interaction en s'appuyant sur un espace unique équivalent associé à chaque menu. Mais il apparaît que des aberrations peuvent se produire : pour sélectionner un menu situé sur la droite de l'axe 0°, une inclinaison de RPM vers la gauche est requise (par exemple le menu grisé dans la Figure 3 – ligne 2, à droite). Pour éviter ce type d'aberrations, il convient donc de prendre en compte la position des items à sélectionner par rapport à l'axe 0° de RPM.

c) Secteurs de taille fixe par sous-plage : La plage d'inclinaisons est divisée en deux sous-plages, à droite et à gauche de l'axe 0° (Figure 3 – ligne 3). Chaque sous-plage

permet de sélectionner les menus situés d'un côté de l'axe 0°. Chaque sous-plage est divisée en autant de secteurs que de menus situés de part et d'autre de l'axe 0°. La taille des secteurs de la sous-plage gauche pourra donc différer de la taille des secteurs de la sous-plage droite selon le nombre de menus situés de chaque côté.

3.3.3 Méthode de validation. Le dernier facteur de conception considéré concerne la méthode de validation. Nous avons anticipé deux techniques différentes: validation par bouton ou par inclinaison.

Avec la validation par bouton, dès que l'utilisateur a réalisé une inclinaison d'une amplitude minimale dans un secteur donné, il peut utiliser le bouton pour valider la sélection. Etant donné que des inclinaisons involontaires de 12° en moyenne ont été mesurées lors de l'utilisation de RPM [27], ce n'est qu'à partir de cette amplitude d'inclinaison qu'un secteur peut être sélectionné.

Avec la validation par inclinaison, l'utilisateur doit réaliser une inclinaison d'une amplitude minimale élevée pour sélectionner un menu. Le seuil d'inclinaison minimal a été fixé à 37 degrés car l'amplitude confortable maximale mesurée avec RPM auprès des utilisateurs est d'environ 37 degrés [27].

3.4 Techniques implémentées

Nous détaillons dans cette section les variations de Rolling-Menu implémentées ainsi que le feedback graphique.

3.4.1 Choix des techniques implémentées. Parmi les variations de Rolling-Menus mises en évidence avec les trois facteurs de conception, nous avons d'abord écarté les versions provoquant des aberrations, i.e. les versions basées sur des secteurs de taille fixe sur l'ensemble de la plage (Figure 3-ligne 2) : nos tests informels ont mis en évidence que ces aberrations sont perturbantes pour l'utilisateur. Nous avons également écarté la version basée sur des secteurs de taille fixe par sous-plage pour une plage directe (Figure 3- ligne 3, gauche), car certains secteurs sont trop petits pour être sélectionnables.

Nous avons ensuite choisi d'implémenter et de comparer deux solutions couvrant des caractéristiques distinctes : Rolling-Menu Direct (Figure 3-ligne 1, gauche) et Rolling-Menu 180 (Figure 3-ligne 3, droite). Nous avons privilégié ces deux solutions car cela permet de comparer l'usage de deux types de plage et de deux types de taille de secteurs différentes.

a) Rolling-Menu Direct (RMDirect) s'appuie sur une plage d'inclinaisons directe et une taille des secteurs variable. RMDirect offre donc un mapping direct entre l'inclinaison à produire sur RPM et la position du menu à activer par rapport à celle de l'activation. (cf. Figure 3-ligne 1, gauche)

b) Rolling-Menu 180 (RM180) fait la distinction entre plage gauche et plage droite. Elle s'appuie sur une plage d'inclinaison de 180° (variant de -90° à +90°) et une taille de secteur fixe par partie (cf. Figure 3-ligne 3, droite). Le secteur situé sur l'axe 0° est de taille fixe. Nous avons établi empiriquement qu'un secteur de 20° était adapté pour une activation aisée et minimisant les erreurs. Cette mesure est compatible avec les résultats de l'étude de la précision des inclinaisons dans 8 directions avec RPM [27] : 8° à 17° de déviation sont observées en fonction des directions.

3.4.2 Feedback graphique. Lorsque le mode Rolling-Menu est activé, des traits continus représentent les secteurs et la plage d'inclinaison (Figure 4). Un point coloré est également

représenté dans la direction de l'inclinaison de RPM : la distance entre l'origine du secteur et ce point représente l'amplitude d'inclinaison de RPM. Une fois le seuil d'inclinaison minimal (12°) dépassé, un feedback d'inclinaison est produit. Il s'agit d'une coloration du secteur dans lequel est incliné RPM :

- Pour un Rolling-Menu avec validation par bouton, le secteur correspondant à l'inclinaison de RPM est coloré en bleu (Figure 4 – droite).
- Pour un Rolling-Menu avec validation par inclinaison, le secteur est tout d'abord coloré en bleu tant que l'amplitude est comprise entre 12° et 30° (Figure 4 - gauche). Dans cette première zone (*zone de sélection*), le secteur sélectionné peut encore être modifié en modifiant la direction d'inclinaison de RPM. Entre 30° et 37° d'amplitude d'inclinaison de RPM, le secteur correspondant est coloré en vert (Figure 4 – centre). Cette zone intermédiaire (*zone de validation*) ignore les changements de direction de l'inclinaison de RPM et constitue un tremplin vers la validation définitive du menu dès que l'amplitude dépasse 37° (cf. section 3.3.3). Elle confère ainsi une plus grande robustesse à la validation.

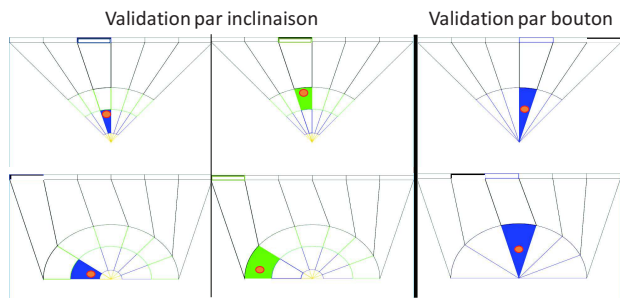


Figure 4: Feedback graphique de Rolling-Menu (RMDirect en haut, RM180 en bas) avec validation par inclinaison (gauche et centre) ou par bouton (droite).

4 ROLLING-MENU VS. SOURIS

Le but de cette étude est de comparer la performance des deux techniques de Rolling-menus implémentées avec celle de la souris classique. Pour chaque technique de Rolling-Menu, nous étudions deux modes de validation : par inclinaison ou par bouton.

4.1 Tâche

La tâche consiste à sélectionner un menu d'une barre de menus horizontale. Nous considérons différents nombres de menus : 4, 6, 8 et 10. La taille des zones sélectionnables correspondant à chaque menu est directement inspirée des menus sur les barres de menus Windows : 25px de hauteur et de 90px (pour 10 menus) à 225px (pour 4 menus) de largeur. Pour toutes les techniques, nous avons contrôlé la position horizontale initiale du pointeur, qui a une influence probable sur la performance. Nous avons considéré trois zones de départ possibles matérialisées par un cercle de 100px de diamètre (cf. Figure 5), au centre, à gauche ou à droite de l'interface. Le centre du cercle était positionné à 550 pixels sous la barre de menus, ce qui correspond à une distance verticale moyenne pour un écran offrant une résolution classique de 1920x1200.

Pour chaque tâche, seule une des trois zones était affichée. Avec la souris, l'utilisateur devait d'abord cliquer dans cette zone, puis sur l'item coloré de la barre de menus (la cible), et enfin revenir cliquer dans la zone initiale. Cette tâche d'aller-retour simule l'interruption provoquée lorsque l'utilisateur doit aller sélectionner un menu de la barre de menus alors qu'il est en train d'interagir avec le contenu d'une application. La tâche de pointage nécessaire pour aller sélectionner le menu dans la barre de menus avait un indice de difficulté compris entre 4.56 et 5.29 selon la position de départ et l'item à sélectionner. L'indice de difficulté de la tâche de pointage pour revenir à la zone de départ était compris entre 2,73 et 3,39.

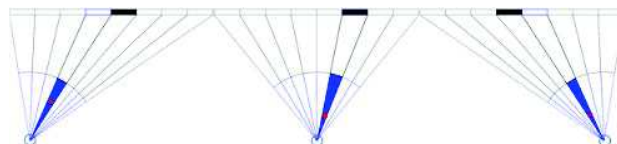


Figure 5: Illustrations pour RMDirect avec 8 menus, des 3 points de départ possibles de la tâche (gauche, centre, droite) dans le protocole expérimental.

Avec les Rolling-Menus, l'utilisateur devait commencer par appuyer sur la barre espace, puis sélectionner l'item de la barre de menus au moyen du Rolling-Menu; pour une validation par bouton, le participant devait enfin appuyer sur la barre espace pour valider sa sélection; pour une validation par inclinaison, le participant devait incliner RPM au-delà du seuil de validation fixé. Afin d'étudier plus particulièrement la performance des inclinaisons, nous avons décidé de ne pas utiliser la translation de la RPM dans l'étude.

L'utilisation d'un dispositif externe pour les validations peut avoir un effet préjudiciable sur la technique. Toutefois l'intégration de capteurs additionnels est technologiquement difficile et nous avons choisi à ce stade de l'étude de ne pas bruyter l'expérience avec de tels problèmes (fils, poids, équilibre, communication, encombrement). Des solutions technologiques sont envisagées en perspectives. Par ailleurs, la tâche d'activation permet d'isoler le temps de pointage du temps de recherche visuelle de la cible à atteindre ; l'utilisation d'une touche de clavier non intégrée au dispositif n'a donc a priori pas d'influence sur la tâche de pointage ; dans une utilisation courante, une simple inclinaison de RPM suffirait à activer le mode menu.

Enfin, nous avons donné comme consigne aux participants de réaliser la tâche aussi vite que possible tout en commettant le moins d'erreurs possible. Un feedback coloré indiquait à l'utilisateur s'il avait réalisé la tâche correctement (vert si correct, rouge si incorrect). Si l'utilisateur commettait une erreur, il n'avait pas à la corriger et devait continuer sa tâche.

4.2 Matériel

L'interface graphique était affichée sur un écran de 17 pouces avec une résolution de 1280px sur 1024px. La fenêtre était affichée sur 900px de large et était centrée à l'écran. L'utilisateur était assis et les dispositifs (souris ou RPM) placés sur une table devant lui, à une hauteur confortable. Les

participants pouvaient positionner comme ils le souhaitent les dispositifs utilisés sur la table.

Nous avons utilisé une RPM de 8 cm de diamètre (cf. Figure 6). La détection des gestes physiques se fait avec une centrale inertielle (xIMU de X-IO technologies). Pour la technique de validation avec bouton, l'utilisateur utilise la barre d'espace du clavier avec la main non-dominante. Pour la souris, nous avons utilisé le gain par défaut de Windows.

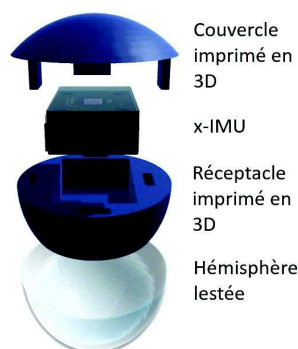


Figure 6: Composants de la Roly-Poly Mouse

4.3 Procédure

L'étude suit un plan 5x4x3 intra-sujets avec comme facteurs la technique d'interaction (RMDirect-Bouton, RMDirect-Inclinaison, RM180-Bouton, RM180-Inclinaison et Souris), le nombre de menus (4, 6, 8 et 10) et la position de départ (Gauche, Centre et Droite). Chaque participant réalise donc un total de $5 \times (4+6+8+10) \times 3 = 420$ essais.

Tous les utilisateurs ont commencé l'expérimentation en utilisant la souris, dans la mesure où sélectionner un menu dans une barre de menus avec une souris est une tâche très classique, maîtrisée par tous. L'ordre de passage des quatre versions de Rolling-Menu était ensuite contrebalancé. Pour les cinq techniques, une période d'entraînement était proposée aux participants, jusqu'à ce qu'ils se sentent à l'aise avec la manipulation de la technique. La durée moyenne de l'expérimentation est de 1 heure.

4.4 Participants

Nous avons recruté 12 participants (6 femmes, 6 hommes) âgés de 29,4 ans en moyenne (SD =9,5). Aucun d'entre eux n'a l'habitude d'utiliser des dispositifs possédant plusieurs DDLs mais six d'entre eux avaient déjà été impliqué dans des expérimentations utilisant RPM. Cinq d'entre eux étaient gauchers.

4.5 Données collectées

Pour chaque essai, nous enregistrons l'ensemble des actions effectuées par les participants (i.e. l'ensemble des événements émis par les dispositifs de pointage utilisés : translations, inclinaisons, rotations, et les pressions et relâchements sur les boutons ou touches), et leur conséquence sur la tâche en cours (i.e. sélection et validation d'un item ou d'une zone de départ). Le temps total mis pour réaliser la tâche ainsi que la sélection correcte ou erronée du menu et les différents secteurs visités sont également enregistrés.

A la fin de chaque bloc associé à une technique d'interaction, les utilisateurs devaient remplir un questionnaire SUS et pouvaient fournir des remarques informelles sur la technique. Une fois l'ensemble des blocs terminés, il leur était demandé de trier les techniques d'interaction selon leur préférence.

5 RÉSULTATS

Nous présentons dans cette section les résultats quantitatifs portant sur le temps et la précision de sélection d'un menu. Nous les complétons par une analyse qualitative qui s'appuie sur l'analyse du score SUS pour chaque technique et l'ordre de préférence exprimé par les participants.

Dans le domaine de l'IHM, les données expérimentales sont traditionnellement analysées par le biais du test de significativité de l'hypothèse nulle. Toutefois, des jugements de plus en plus critiques à l'égard de ces approches et de leurs mises en œuvre dans de telles situations expérimentales, sont émis par les communautés de l'IHM [13,14] et de la Statistique [5,12]. Pour analyser nos résultats, nous nous appuyons donc sur des techniques d'estimations basées sur des amplitudes d'effet et des intervalles de confiance à 95%, en accord avec les récentes recommandations de l'APA [32]. Les scripts utilisés sont ceux utilisés par [33] et disponibles en ligne (<http://www.aviz.fr/reliefshearing>, section "study material")

5.1 Temps et précision de sélection d'un menu

L'analyse des temps mis pour réaliser la sélection d'un menu avec chacune des cinq techniques d'interaction et quels que soit le nombre de menus affichés, montre qu'en moyenne l'utilisation des Rolling-Menus avec validation par bouton permet de sélectionner un menu plus rapidement qu'avec une souris (cf. Figure 7).

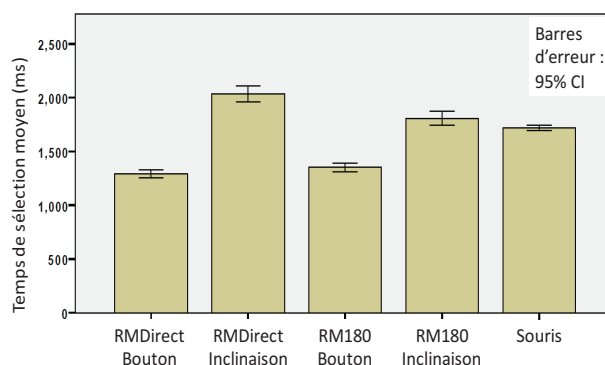


Figure 7: Temps de sélection moyen (ms) par technique d'interaction.

L'analyse de ces résultats basée sur le calcul des ratios du temps de sélection avec la souris sur le temps de sélection avec les Rolling-Menu confirme que cette différence est nettement établie : il apparaît en effet qu'avec la souris les participants mettent en moyenne 34,0% (CI [24.3%, 43.4%]) de temps en plus qu'avec RMDirect-Bouton, et 32,1% (CI [18.2%, 45.4%]) de temps en plus qu'avec RM180-Bouton. Il ressort également que la souris est aussi rapide que RM180-Inclinaison, et est 14,2% (CI

[7.9%, 22.8%]) plus rapide que RMDirect-Inclinaison (cf. Figure 8).

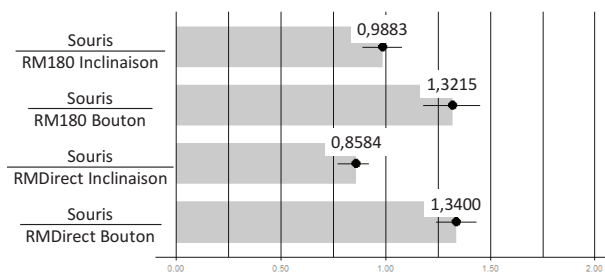


Figure 8: Ratio des temps de sélection et intervalles de confiance à 95%. Les ratios représentent le temps mis avec la souris (Ts) par rapport au temps mis par chaque version de Rolling-Menu (Trm). Pour un ratio >1, cela signifie donc que Ts > Trm. Les temps pris en compte sont les moyennes géométriques des temps mis par chaque utilisateur avec chaque technique.

Cette première analyse ne tient pas compte du nombre de menus contenus dans la barre de menus. Pourtant, en prenant en compte ce facteur, il apparaît que la sélection avec les Rolling-Menu est systématiquement plus rapide, quelle que soit la méthode de validation pour une barre de menus contenant 4 ou 6 menus. Il ressort également que l'utilisation d'un Rolling-Menu avec validation par bouton demeure plus rapide que l'utilisation de la souris, quel que soit le nombre de menus contenus dans la barre de menus (cf. Figure 9).

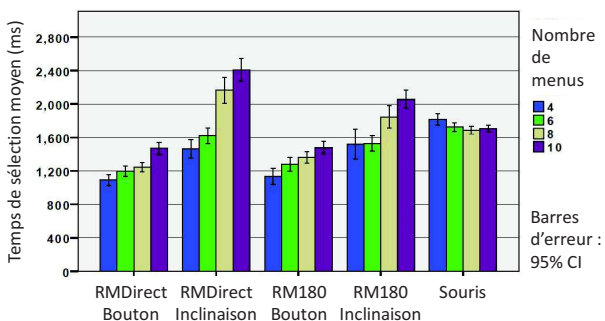


Figure 9: Temps de sélection moyen (ms) par technique d'interaction et par nombre de menus affichés dans la barre de menus.

Ces constatations sont confirmées par l'analyse des ratios des temps de sélection avec les Rolling-Menu par rapport à la souris.

Quand le nombre de menus sélectionnables est strictement inférieur à 8, il apparaît en effet que la souris est systématiquement et nettement plus lente (entre 14,7% et 55,6%) que les différentes versions de Rolling-Menu (cf. Figure 10).

Quand le nombre de menus sélectionnables est égal à 8 ou 10, cette analyse des ratios établie également avec netteté que la souris est moins rapide que les Rolling-Menus avec validation par Bouton : la souris est plus lente de 22,8% (CI [10,6%, 34,2%]) que RM180-Bouton et de 25,0% (CI [14,9%, 33,0%]) que RMDirect-Bouton (cf. Figure 11).

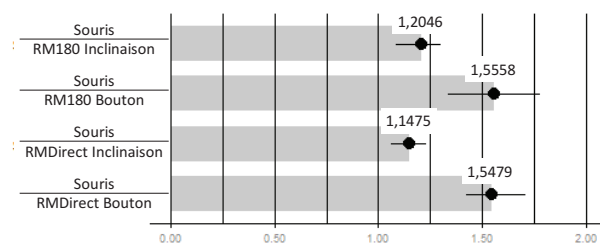


Figure 10: Ratio des temps de sélection et intervalles de confiance à 95% pour des barres de menus contenant 4 ou 6 menus. Si ratio > 1, alors la souris est plus lente.

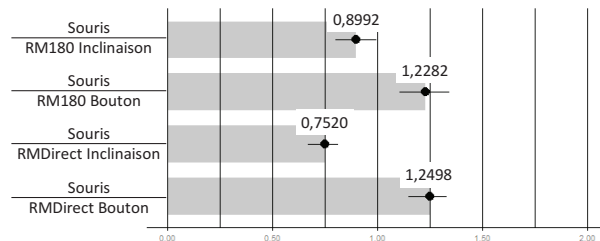


Figure 11: Ratio des temps de sélection et intervalles de confiance à 95% pour des barres de menus contenant 8 ou 10 menus. Si ratio > 1, alors la souris est plus lente.

En termes de précision, les résultats obtenus avec les cinq techniques sont très similaires et atteignent 95,9% toutes techniques confondues (Souris : 96,7% (CI [94,5%, 97,9%]), RM180-Inclinaison : 93,9% (CI [90,4%, 96,1%]), RM180-Bouton : 97,5% (CI [96,0%, 98,5%]), RMDirect-Inclinaison 94,1% (CI [91,3%, 96,2%]), RMDirect-Bouton : 97,0% (CI [95,3%, 97,9%])). La technique de validation par inclinaison a donc tendance à induire légèrement plus de sélections de menu erronées : le taux de succès de la souris par rapport à RMDirect-Inclinaison est de 102,9% (CI [99,6%, 105,5%]) et il est de 103,3% (CI [99,9%, 107,4%]) par rapport à RM180-Inclinaison.

Enfin, l'amplitude des inclinaisons étaient prédéfinies pour une validation par inclinaison, mais laissée libre pour les validations par bouton : le participant choisissait lui-même le moment auquel valider la sélection, dans la mesure où il dépassait le seuil minimal de 12°. L'analyse des amplitudes d'inclinaison moyennes au moment de la validation par bouton montre que les participants inclinent RPM de 30,9° [CI : 27,0°; 34,7°] en moyenne. Cette inclinaison est en cohérence avec les résultats précédemment mis en évidence [27].

5.2 Analyse qualitative

Le score SUS obtenu en moyenne par la souris (89,2 ; CI[84,0, 94,0]) est supérieur à ceux obtenus par les 4 versions de Rolling-Menu. Toutefois, selon l'équivalence établie entre le score et le qualificatif de la technique [9], RM180-Bouton est aussi jugée excellente (86,5 ; CI[78,8, 91,7]), RMDirect-Bouton (81,9 ; CI[84,0, 94,0]) et RM180-Inclinaison (71,9 ; CI[66,0, 78,1]) sont jugées bonnes et RMDirect-Inclinaison (65,8 ; CI[57,9, 73,1]) est jugée "ok" (cf. Figure 12 - haut).

L'analyse des ratios des scores SUS établit que le score SUS obtenu par la souris est très nettement supérieur à celui obtenu par les Rolling-Menu basés sur une validation par inclinaison (RMDirect-Inclinaison : 41,0%, CI[25,4%, 61,4%]; RM180-

Inclinaison : 27.0%, CI[14.1%, 40.2%]) à ceux obtenus avec la souris (cf. Figure 12 - bas). La différence est plus nuancée pour les Rolling-Menu utilisant une validation par bouton.

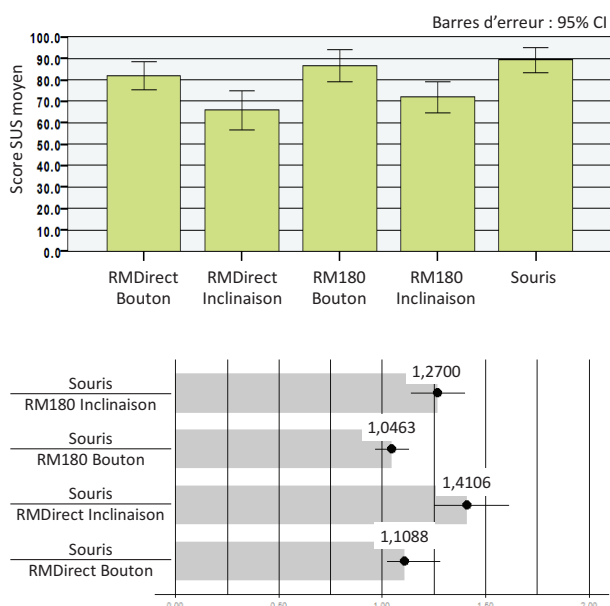


Figure 12: Scores SUS moyens pour les cinq techniques d'interactions (haut) et ratios du score SUS et intervalles de confiance à 95% (bas). Si ratio > 1, alors le SUS de la souris est supérieur.

Pour analyser la préférence des utilisateurs entre ces cinq techniques, nous nous sommes appuyés sur le classement en ordre de préférence établi par chacun des participants à l'issue de leur passation. En cumulant le nombre de fois que chaque technique a été classée au rang 1 ou 2, RM180-Bouton (8 fois) émerge nettement des 4 autres, devant la souris (7 fois), RM180-Inclinaison (5 fois), RMDirect-Bouton (3 fois) et RMDirect-Inclinaison (1 fois). Il est aussi intéressant d'observer que RMDirect-Bouton et RM180-Bouton n'ont jamais été classées en dernière position, alors que la souris a été classée 4 fois sur 12 en dernière position.

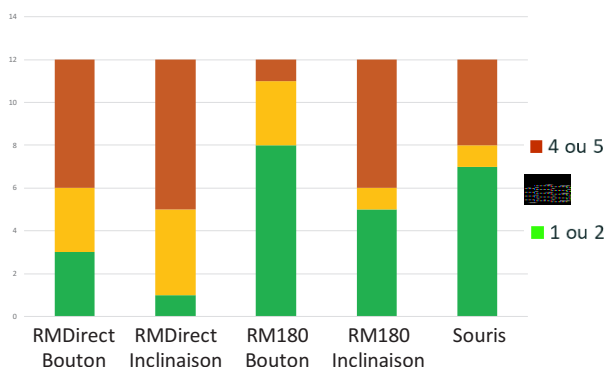


Figure 13: Cumul des ranking par techniques d'interaction.

Enfin au cours de l'expérimentation nous avons recueilli les commentaires positifs et négatifs des participants au sujet des différentes techniques testées. Ils ont en particulier souligné que la petite taille des secteurs avec RMDirect "compliquait l'utilisation de la technique et nécessitait par conséquent une plus grande concentration". RM180 en revanche donnait l'impression à plusieurs participants "d'être plus efficace que la souris". Toutefois "le non alignement entre l'inclinaison attendue et le menu à sélectionner peut paraître perturbant" pour certains utilisateurs. Les participants ont aussi souligné que l'utilisation des Rolling-Menu pouvait se faire en "maintenant le poignet posé sur la table, limitant ainsi la fatigue" et "permettant une interaction décontractée". Enfin concernant le mécanisme de validation, si l'inclinaison est apparue "plus compliquée" et nécessitant "une attention plus longue pour maintenir la direction de l'inclinaison", les participants ont aussi déploré que la validation par bouton nécessite une interaction bimanuelle, non intégrée.

Qualitativement, il ressort donc que RMDirect-Inclinaison est la technique la moins appréciée des utilisateurs et que RM180-Bouton est la plus appréciée.

RMDirect est le Rolling-Menu qui offre les plus petits secteurs pour sélectionner un menu ; de plus la direction de l'inclinaison doit être maintenue dans ce secteur étroit jusqu'à atteindre l'amplitude de validation : ceci demande donc beaucoup d'attention de la part de l'utilisateur et explique le classement en dernière position. A l'inverse, RM180-Bouton offre des secteurs de taille maximale avec une validation possible dès que la direction de l'inclinaison de RPM coïncide le secteur du menu à sélectionner.

6 DISCUSSION

6.1 Apports de Rolling-Menu

Cette évaluation expérimentale établie donc que pour une barre de menus contenant 4 ou 6 menus, toutes les techniques de Rolling-Menu sont plus rapides que la souris pour sélectionner un menu. Pour des barres de menus contenant un plus grand nombre de menus (8 ou 10), les Rolling-Menu basées sur une validation par bouton sont systématiquement plus rapides que la souris. Ces deux résultats sont d'autant plus intéressants qu'ils s'accompagnent d'un taux d'erreurs tout à fait équivalent à celui observé avec la souris.

Du point de vue qualitatif, l'utilisabilité des Rolling-Menu basés sur une validation par bouton (score SUS) est similaire à celle de la souris. En revanche l'utilisabilité des Rolling-Menu avec validation par inclinaisons doit être améliorée.

En cohérence avec ces résultats quantitatifs et qualitatifs, le RM180-Bouton est la technique d'interaction préférée par les participants pour permettre l'activation d'un menu dans une barre de menus.

Plus généralement, Rolling-Menu permet d'activer un menu d'une barre de menus sans déplacer le pointeur de l'application avec laquelle l'utilisateur est en train d'interagir. De nombreuses situations peuvent en tirer profit : les environnements de dessin où il est nécessaire par exemple d'aller chercher l'outil adéquate dans une palette (e.g. la gomme) et de revenir au niveau du dessin pour utiliser cet outil; les tableurs, au cours de l'édition d'une formule de calcul

nécessitant la référence à une cellule que l'utilisateur va cliquer; l'environnement windows dans lequel le recours à la barre des tâches pour sélectionner une application nécessite de repositionner le curseur pour interagir avec le contenu de l'application sélectionnée; ou encore les éditeurs de texte et notamment les menus d'insertion de formes graphiques. Rolling-Menu est donc susceptible d'apporter une réponse efficace à l'interaction avec de nombreuses applications interactives de différents domaines.

Enfin, soulignons que dans ce dernier exemple, l'activation du menu "insertion d'une forme géométrique" ne donne pas lieu à l'ouverture d'un sous menu contenant des menus organisés dans une liste verticale, mais à une bulle contenant une matrice de 28 x 12 items différents sélectionnables. De par ses multiples degrés de liberté, RPM, et donc un Rolling-Menu, permet d'envisager un déplacement optimisé dans un tel sous-menu. Rolling-Menu est donc compatible avec des menus plus riches que des hiérarchies de menus linéaires horizontales et verticales.

6.2 Limite de Rolling-Menu

Dans les versions actuellement évaluées de Rolling-Menu, la principale limite reste l'absence d'un bouton intégré pour la validation. Le recours à un bouton externe établit pourtant l'intérêt de ce dispositif de validation. Trois solutions technologiques sont toutefois envisageables à très court terme : un bouton circulaire a déjà été prototypé ([27] – Figure 11); une version plus récente de RPM intègre un écran tactile permettant d'envisager un système de validation sur le dessus de RPM [30]; en plaçant RPM sur un tapis de souris capacitif et donc tactile (e.g. tablette Wacom Intuos Pro), il devient possible de déclencher un "clic" avec un doigt sur le tapis (des tests informels indiquent que cela est possible de façon confortable avec le pouce). Ces solutions intégrées sont prometteuses et réalistes mais devront toutefois faire l'objet d'une étude approfondie.

Au-delà du mécanisme de validation, RPM souffre d'une autre limitation technologique. La centrale inertielle embarquée à ce jour dispose d'une autonomie relativement faible. Elle est aussi sujette à une dérive qui empêche d'envisager un fonctionnement correct du Rolling-Menu au-delà d'une dizaine de minutes.

Plus spécifique au Rolling-Menu lui-même, et plus particulièrement aux versions basées sur une validation par inclinaison, il semble nécessaire de mieux ajuster les seuils de passage entre zone de sélection, zone de validation et validation définitive. Il ressort en effet que l'utilisation d'un Rolling-Menu avec validation par Bouton induit des amplitudes d'inclinaisons inférieures (env. 15%) au seuil de 37° fixé conformément aux premières études [27]. Réduire l'inclinaison requise pour la validation par inclinaison pourrait aussi réduire le temps de sélection. Une étude est toutefois requise pour mesurer l'impact de la réduction de cette amplitude sur la souplesse de sélection du secteur.

Ces limitations technologiques ont aussi eu un impact sur l'expérimentation elle-même. En effet nous n'avons conduit qu'une étude contrôlée en laboratoire sur une tâche expérimentale et donc abstraite. En intégrant l'usage de Rolling-Menu dans un environnement interactif réel, les modes d'utilisation de Rolling-Menu pourraient être mieux cernés. En

particulier, l'expérimentation a contraint le point de départ de la tâche en trois lieux de l'écran (gauche, centre, droit). Nous pensons toutefois, qu'après avoir activé le mode menu et donc figé le pointeur de l'application, les utilisateurs pourraient vouloir déplacer l'origine des secteurs de Rolling-Menu pour donner au secteur visé la taille la plus grande possible et ainsi terminer par une inclinaison (probablement le long de l'angle 0°) rapide et simple.

Enfin en termes de tâches couvertes par l'expérimentation, l'étude s'est focalisée sur la sélection d'un menu d'une barre de menus contenant au plus 10 menus. Notre étude n'a pas considéré un nombre plus grands de menus, ni la tâche de sélection d'un sous-item ou d'un menu dans un menu hiérarchique. Il semble néanmoins que le geste d'inclinaison en sens inverse, lors du retour de RPM à son équilibre, constitue une approche prometteuse pour permettre la navigation dans les sous-menus ouverts par l'activation du menu initial. Cette solution reste compatible avec le but initial de Rolling-Menu qui vise à laisser le curseur de l'application à la place où il se trouve lors de l'activation du mode menu.

6.3 Perspectives

Nous identifions trois perspectives complémentaires à l'étude que nous avons conduite et aux limites déjà identifiées.

La première consiste à envisager la conception d'un mode d'utilisation expert de Rolling-Menu. Dans les versions testées, il s'agissait d'un mode novice, dispensant un feedback directement sur l'application (éventuellement en transparence). Le mode expert pourrait s'abstraire de ce feedback ou le déporter sur un écran périphérique tels que celui ajouté à la RPM dans TDome [30]. Le mode expert pourrait aussi mieux tirer profit de la « mémoire musculaire » de l'utilisateur. A ces fins, il pourra s'avérer utile de considérer l'origine du menu en un point unique de l'écran : la sélection d'un menu serait alors associée à un mouvement physique toujours identique, quelle que soit la position initiale du curseur. De nouvelles considérations de conception se posent toutefois dans la mesure où une barre de menus est rarement répartie de façon homogène sur toute la largeur de la fenêtre active et que la distance entre l'objet de travail et l'origine du menu induit une nouvelle transition susceptible d'impacter la performance d'utilisation du Rolling-Menu.

La seconde concerne le nombre de menus contrôlables par Rolling-Menu. Les résultats de l'expérimentation soulignent notamment que jusqu'à 6 menus, toutes les versions de Rolling-Menu permettent d'interagir avec une barre de menus plus rapidement qu'avec la souris. Toutefois 6 menus sont rarement suffisants dans une application. S'appuyant sur ce double constat, une perspective d'évolution des Rolling-Menu consiste à associer deux menus à un même secteur. Puis la rotation de RPM (autour de son axe central) permettrait d'une part de distinguer le menu à sélectionner parmi les deux associés au secteur sélectionné et, d'autre part, de déclencher la validation de la sélection. Cette nouvelle version de Rolling-Menu éviterait ainsi la validation par inclinaison qui semble peu convaincante, ne nécessiterait pas l'ajout d'un bouton sur RPM et multiplierait par 2 le nombre de menus accessibles par toute version de Rolling-Menu (8 à 12 menus). Une expérimentation devra bien entendu valider la pertinence de ce nouveau design.

Enfin, deux contextes d'utilisation de Rolling-Menu sont envisageables : 1) la sélection d'un menu dans une barre de menus et ses sous-menus hiérarchiques, 2) la sélection d'une icône dans une barre d'outils, d'icônes ou dans un « ribbon » similaire à ceux affichés dans la suite Office. Dans ces deux contextes, des travaux complémentaires sont nécessaires : d'une part, la sélection d'un item du sous-menu (pull-down-menu) ouvert et la navigation dans un menu hiérarchique sont des tâches pour lesquelles nous devons encore explorer les différentes solutions d'interaction possibles avec Rolling-Menu; d'autre part, la sélection d'une icône de la barre d'outils (ou du ribbon activé) amène des contraintes complémentaires telles que la hauteur finie des icônes, l'accroissement du nombre de cibles possibles, la variabilité de la disposition des cibles. La performance et les limites de Rolling-Menu dans ces deux contextes devront donner lieu à de nouvelles expérimentations et mèneront à comprendre dans quelle mesure les Rolling-Menu peuvent être associés à d'autres types d'utilisation tels que menu contextuel, circulaires ou verticaux.

7 CONCLUSIONS

Cet article a présenté Rolling-Menu, une technique d'interaction basée sur les inclinaisons de la Roly-Poly Mouse, pour interagir avec une barre de menus. Rolling-Menu offre une meilleure intégration entre la sélection de menus et la manipulation directe du contenu d'une application. Nous avons proposé un espace de conception pour la conception de Rolling-Menu et avons implémenté quatre versions différentes. Une étude expérimentale a établi les apports de Rolling-Menu en termes de vitesse, précision, utilisabilité et préférence. Il ressort en particulier que les Rolling-Menu avec validation par bouton sont systématiquement plus rapides que la souris pour sélectionner un menu. RM180-Bouton est d'ailleurs la technique préférée des participants et jusqu'à 37% plus rapide que la souris.

REMERCIEMENTS

Les auteurs remercient Mathias Faure pour sa contribution à la conception et à l'implémentation des premières techniques d'interaction Rolling-Menu. Ces travaux sont partiellement financés par le projet ANR AP2 ANR-15-CE23-0001.

RÉFÉRENCES

- Ahlstrom, D. and David. Modeling and improving selection in cascading pull-down menus using Fitts' law, the steering law and force fields. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*, ACM Press (2005), 61–70.
- Bailly, G. Techniques de menus: caractérisation, conception et évaluation. <http://www.theses.fr>, 2009. <http://www.theses.fr/2009GRE10062>.
- Bailly, G., Demeure, A., Lecolinet, E., and Nigay, L. MultiTouch menu (MTM). *Proc. of the 20th International Conference IHM '08*, ACM Press (2008), 165.
- Bailly, G., Lecolinet, E., and Nigay, L. Flower menus: a new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. *Proceedings of the working conference on Advanced visual interfaces - AVI '08*, (2008), 15.
- Baker, M. Statisticians issue warning on Pvalues. *Nature*, (2016).
- Balakrishnan, R., Baude, T., Kurtenbach, G., and Fitzmaurice, G. The Rockin'Mouse: Integral 3D Manipulation on a Plane. *Proceedings ACM SIGCHI Conference on Human Factors in Computing Systems*, (1997), 311–318.
- Benko, H., Izadi, S., Andrew D. Wilson, Cao, X., Rosenfeld, D., and Hinckley, K. Design and evaluation of interaction models for multi-touch mice. *Proceedings of Graphics Interface 2010*, A K Peters (2010), 276.
- Blanch, R., Guiard, Y., and Beaudouin-Lafon, M. Semantic Pointing - Improving Target Acquisition with Control-display Ratio Adaptation. *Proceedings of the International Conference on Human Factors in Computing*

- Systems (CHI'04)* 6, 1 (2004), 519–526.
- Brooke, J. SUS: A quick and dirty usability scale. *Usability Evaluation in Industry*, (1996), 189–194.
- Callahan, J., Hopkins, D., Weiser, M., and Shneiderman, B. An Empirical Comparison of Pie vs. Linear Menus. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (1988), 95–100.
- Cockburn, A., Gutwin, C., and Greenberg, S. A predictive model of menu performance. *Proc. CHI 2007*, (2007), 627–636.
- Cumming, G. The new statistics: Why and how. *Psychological Science*, (2014).
- Dragicevic, P. Fair Statistical Communication in HCI. Springer, 2016, 291–330.
- Dragicevic, P., Chevalier, F., and Huot, S. Running an HCI experiment in multiple parallel universes. *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems - CHI EA '14*, ACM Press (2014), 607–618.
- Grossman, T., Hinckley, K., Baudisch, P., Agrawala, M., and Balakrishnan, R. Hover Widgets: Using the Tracking State to Extend the Capabilities of Pen-Operated Devices. *CHI 2006*, ACM Press (2006), 861–870.
- Guimbretière, F. and Nguyen, C. Bimanual marking menu for near surface interactions. *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, ACM Press (2012), 825.
- Guimbretière, F. and Winograd, T. FlowMenu - Combining Command, Text, and Data Entry. *Proceedings of the 13th International Symposium on User Interface Software and Technology (UIST'00)* 2, (2000), 213–216.
- Hascoët, M., Collomb, M., and Cance, J. Accelerating object-command transitions with pie menus. *Enactive'06: Conference on Enaction and Complexity*, (2006), 109–111.
- Hinckley, K. and Sinclair, M. Touch-sensing input devices. *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, ACM Press (1999), 223–230.
- Hinckley, K., Sinclair, M., Hanson, E., Szeliski, R., and Conway, M. The VideoMouse: a camera-based multi-degree-of-freedom input device. *Proceedings of the 12th annual ACM symposium on User interface software and technology - UIST '99*, ACM Press (1999), 103–112.
- Huot, S., Nancel, M., and Beaudouin-Lafon, M. *PushMenu: Extending Marking Menus for Pressure-Enabled Input Devices*, Research Report LRI-RR1502. Paris, 2008.
- Kim, S., Kim, H., Lee, B., Nam, T.-J., and Lee, W. Inflatable mouse: volume-adjustable mouse with air-pressure-sensitive input and haptic feedback. *Proceeding of the twenty-sixth annual [SIGCHI] conference on Human factors in computing systems*, (2008).
- Kobayashi, M. and Igarashi, T. Considering the direction of cursor movement for efficient traversal of cascading menus. *Proc. of the 16th annual ACM symposium on User interface software and technology - UIST '03*, ACM Press (2003), 91–94.
- Kurtenbach, G., Sellen, A., and Buxton, W. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Human-Computer Interaction* 8, 1 (1993), 1–23.
- Lepinski, G.J., Grossman, T., and Fitzmaurice, G. The design and evaluation of multitouch marking menus. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, ACM Press (2010), 2233.
- Perelman, G., Serrano, M., Raynal, M., Dubois, E., Picard, C., and Derras, M. Designing an input device to interact with multidimensional data: disco. *Proceedings of the 26th Conference on l'Interaction Homme-Machine - IHM '14*, ACM Press (2014), 91–100.
- Perelman, G., Serrano, M., Raynal, M., Picard, C., Derras, M., and Dubois, E. The Roly-Poly Mouse: Designing a Rolling Input Device Unifying 2D and 3D Interaction. *SIGCHI Conference on Human Factors in Computing Systems - CHI 2015*, ACM SIGCHI (2015), 10 pages (to appear).
- Roudaut, A., Lecolinet, E., and Guiard, Y. MicroRolls: Expanding Touch-Screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb. *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, ACM Press (2009), 927.
- Roy, Q., Malacria, S., Guiard, Y., Lecolinet, E., and Eagan, J. Augmented letters. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, ACM Press (2013), 2325.
- Saidi, H., Serrano, M., Irani, P., and Dubois, E. TDome: A Touch-Enabled 6DOF Interactive Device for Multi-Display Environments. *Proceedings of CHI'17*, (2017), to appear.
- Tian, F., Xu, L., Wang, H., et al. Tilt Menu: Using the 3D Orientation Information of Pen Devices to Extend the Selection Capability of Pen-based User Interfaces. *Proceedings of ACM CHI 2008 Conference on Human Factors in Computing Systems*, (2008), 1371–1380.
- VandenBos, G.R., ed. *Publication Manual of the American Psychological Association (6th ed.)*. American Psychological Association, 2009.
- Willett, W., Jenny, B., Isenberg, T., et al. Lightweight Relief Shearing for Enhanced Terrain Perception on Interactive Maps. *Proceedings of the 33rd ACM Conference on Human Factors in Computing Systems (CHI 2015)*, (2015).
- Yang, X.-D., Mak, E., McCallum, D., Irani, P., Cao, X., and Izadi, S. LensMouse: augmenting the mouse with an interactive touch display. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, ACM Press (2010), 2431.
- Zhao, S. and Balakrishnan, R. Simple vs. compound mark hierarchical marking menus. *Proceedings of the 17th annual ACM symposium on User interface software and technology - UIST '04*, ACM Press (2004), 33.
- The Z: The most advanced gaming mouse ever made. 2017. <https://www.kickstarter.com/projects/granto/the-z>.