

ClavPad

Contexte et objectifs:

Tout ordinateur portable est équipé d'un clavier et d'un touchpad. Pour certaines tâches, les utilisateurs préfèrent utiliser une souris qui possède de nombreux avantages par rapport à un touchpad, notamment la précision...

Malgré cela, le touchpad est beaucoup utilisé pour plusieurs raisons: il permet de réduire le homing entre le clavier et la souris, il permet d'effectuer diverses actions grâce à des raccourcis à plusieurs doigts, et il est toujours présent sur un ordinateur portable (pas besoin de transporter une souris supplémentaire) !

Cependant, il existe toujours un homing entre le clavier et le touchpad qui peut être limitant dans des tâches où les allers-retours touchpad/clavier sont récurrents (édition de texte ...). De plus la zone d'un touchpad est assez limitée.

Notre principal objectif est donc de réduire au maximum ce homing.

Solution envisagée:

Notre projet consiste à rendre le clavier tactile afin de déplacer le curseur de la souris (indirection) et effectuer certaines actions comme le clic gauche. Cela permettra de réduire au maximum le homing et d'agrandir la zone "tactile" du touchpad.

Nous allons nous concentrer sur un ClavPad en "position absolue" pour le déplacement du curseur. Puisque nous nous intéressons particulièrement à la réduction du homing et non pas à la vitesse ou à la précision du système.

Fonctionnement du ClavPad:

Pour passer en mode "tactile", il suffira de serrer deux doigts (l'index et le majeur) et de toucher le ClavPad.

Pour bouger le curseur, il faudra faire glisser les deux doigts serrés dans la direction souhaitée sur la ClavPad.

Pour effectuer le clic gauche, il faudra "appuyer" avec les deux doigts serrés sur le ClavPad sans enfoncer les touches du clavier. Cela se rapproche donc plus d'un "toucher tactile".

Problèmes rencontrés et solutions mises en place:

Problème système optitrack, écart des doigts:

En fonction de la morphologie des doigts, la distance entre les 2 marqueurs en position "tactile" (2 doigts serrés pour utiliser le système ClavPad) est supérieure à la distance entre les 2 marqueurs lorsque l'on écrit sur le clavier en utilisant 2 lettres diagonalement à côté (par exemple k et o).

Notre solution est donc de ne pas forcément mettre les 2 marqueurs sur le bout des ongles des doigts mais le plus proche possible lorsque les 2 doigts sont en position serré (ex: un marqueur sur le bout de l'ongle, l'autre au début de l'ongle voir sur la peau).

Problème ratio écran / clavier:

Contrôle du curseur au niveau du bord de l'écran. En effet, le clavier entier étant projeté sur la taille de l'écran, la sensibilité sur l'axe X et Y n'est plus identique. Du plus, lorsque l'on sort de la zone du clavier le contrôle du curseur s'arrête jusqu'au retour dans celle-ci.

Une solution consiste à réduire la zone de contrôle du clavier en imposant de conserver le format de l'écran. Cela permet aussi de réduire la zone de trajet pour atteindre un point précis. Dans notre cas, nous avons choisi d'utiliser le clavier de l'ordinateur portable qui possède le même format que l'écran.

Problème de précision des clics:

Avec les clics en survol, il était très difficile de rester précis car le mouvement bref qui était nécessaire pour effectuer un clic menait l'utilisateur à faire bouger le curseur pendant l'action. Nous avons donc revu la façon de détecter un clic. En effet, au lieu de mesurer l'accélération nous avons plutôt choisi un seuil sur la hauteur au clavier. (déterminé à la calibration) Lorsqu'en mode curseur la hauteur du premier marqueur passe en dessous du seuil, il s'agit d'un clic et aucun clic supplémentaire n'est considéré tant que l'utilisateur ne quitte pas cette zone.

Problème pour les clics:

On avait parfois un double, triple ou quadruple clic à la place d'un seul clic demandé. La solution précédente a permis de corriger ce problème.

Notre prototype:

Afin de simuler le ClavPad, nous utiliserons le système optitrack qui, avec quatre caméras et des marqueurs, nous permettra de détecter les positions du clavier et des doigts dans l'espace. Cela nous permettra de "créer" notre prototype de clavier tactile (ClavPad) à moindre coût.

Présentation du système optitrack:

Le système optitrack permet de détecter la position d'objets dans le champ des caméras. La détection d'objet est possible à l'aide de marqueur que l'on positionne dans le champ des caméras.

Afin de prototyper le ClavPad, nous avons besoin :

- d'un clavier équipé de marqueur (ici quatre).
- de quatre caméras optitrack positionnées autour d'un clavier.
- de doigts équipés d'un marqueur chacun.
- du logiciel *Motive* qui permet de paramétrer, calibrer et accéder aux données du système optitrack.

Code Clavpad (en C++):

Ce code permet de récupérer les données du système optitrack (position des doigts, du clavier, ...) et de les traiter afin de simuler le ClavPad (clavier tactile) et donc de faire bouger le curseur de la souris.

Il fonctionne comme une machine à état où les états possibles sont:

- IDLE pas d'interaction particulière
- KEYBOARD lorsqu'une touche du clavier est enfoncée
- CURSOR_MOVE lors d'un déplacement de deux doigts accolés
- CURSOR_CLICK lors d'une détection de clic
- MOUSE_MOVE
- MOUSE_CLICK

Les deux derniers ne sont pas utilisés.

Il permet également de nous renvoyer des valeurs, notamment pour nous permettre d'évaluer l'interaction et le homing (nous avons finalement trouvé un outil permettant de mesurer le homing directement: [Homing Experiment Software Instructions \(cornell.edu\)](https://www.cornell.edu/homing-experiment-software-instructions)).

Dans un premier temps, une étape de calibration est nécessaire afin de fixer les constantes liées à l'utilisateur comme la distance entre les doigts pour le contrôle du curseur (seuil sur la distance entre les 2 doigts) où le schéma utilisateur pour un click.

Dans un deuxième temps, le programme reçoit en temps réel les positions des marqueurs du rigid body (clavier) et celles des marqueurs collés aux doigts.

Il calcul l'écart des doigts afin de déterminer s'il faut passer en mode "tactile" (doigts serrés). Si c'est le cas, alors il détermine la position du curseur en fonction de la position des doigts sur le clavier. Ainsi, le curseur se déplace en même temps que les doigts sur le clavier.

Le programme calcul également la hauteur des doigts afin de repérer un clic de la part de l'utilisateur lorsque ceux-ci passe en dessous du seuil d'un clic.

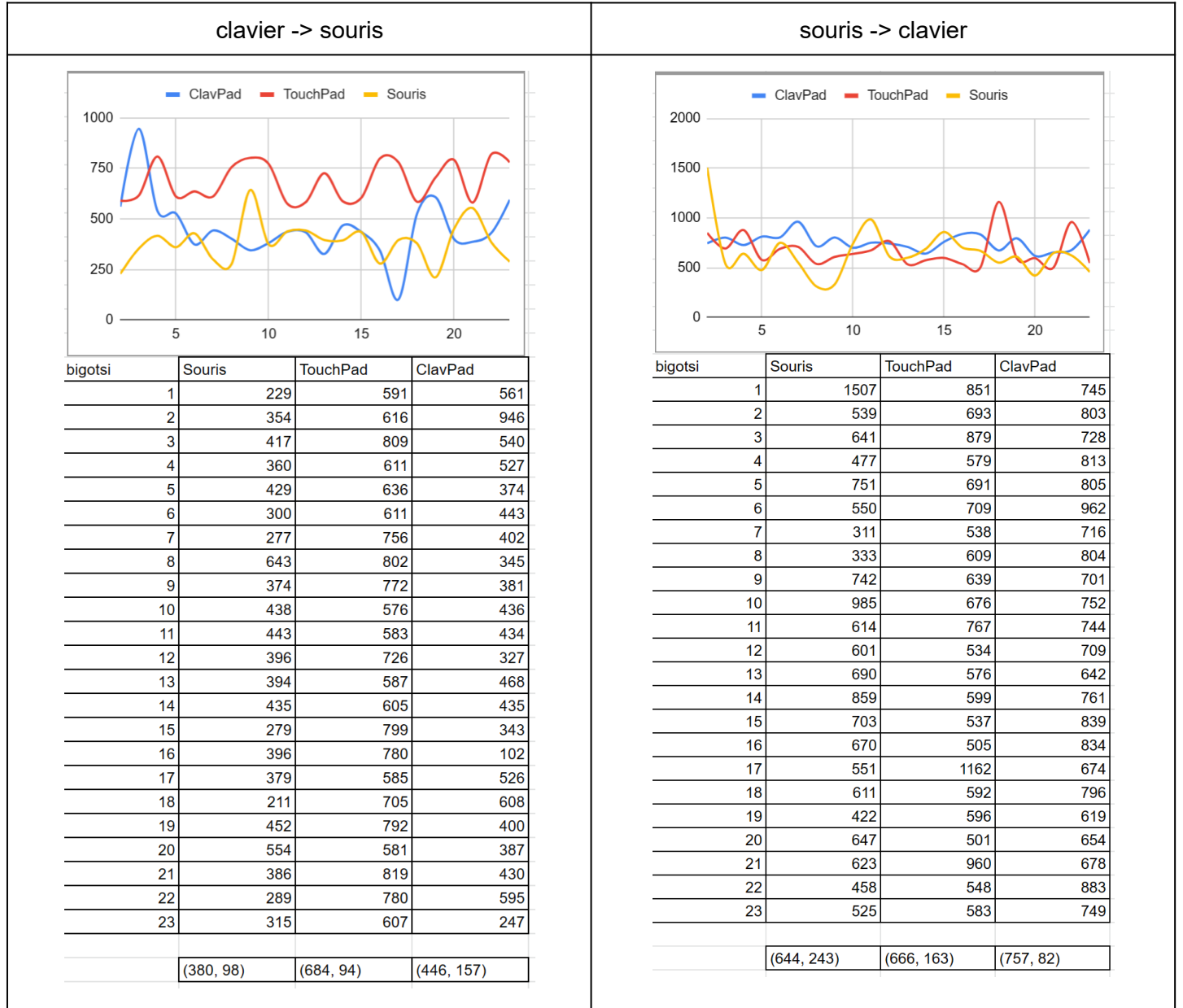
Le programme détecte également l'appui sur les touches du clavier afin de repasser en mode "clavier".

Afin d'évaluer notre interaction, le programme permet également de récupérer la time-line des temps "clavier" et "tactile" ce qui nous permettra de calculer le homing, ...

Evaluation du homing:

Nous n'avons pas pu enregistrer les données de Homing pour les autres membres du groupe. Notamment suite à un changement de code du casier ...

Ci-dessous les résultats de Homing pour un seul membre du groupe :



La dernière ligne indique (moyenne, écart type).

Notre code permettant de passer du mode clavier au mode souris en serrant les doigts demande d'attendre 30 frames. Il faut donc tenir compte d'un petit temps constant qui augmente le Homing de clavier vers souris.

Observations:

Pour le passage du mode clavier au mode tactile, les résultats obtenus à l'aide du ClavPad sont très satisfaisants mais on ne peut pas dire la même chose pour l'inverse (passage du mode tactile au mode clavier) car il n'y a pas de gain (mais pas non plus de perte).

Bien entendu, on ne peut pas attendre beaucoup de ces résultats car ce n'est que pour une personne, avec un seul test pour chaque mode (souris, touchpad et clavpad) ...

Un intérêt de ce dispositif est qu'il permet de positionner pendant le homing puisque le curseur se téléporte à l'endroit voulu se qui se rapproche d'un contrôle intégrale sur le passage de clavier à curseur et positionnement.

Conclusion:

On ne peut pas réellement conclure sur la réduction du homing car nous n'avons pas pu effectuer suffisamment de tests et corriger le délai de 30 frames.

Les avantages d'utiliser le positionnement absolu pour notre clavPad incluent la précision de la localisation de la position de l'utilisateur sur le touchpad, ainsi que la facilité de mappage de cette position à une action ou une commande spécifique au niveau de l'écran.

Si on aurait eu plus de temps de test pour le projet, nous avons pensé à différents points:

- Tout d'abord l'optimisation du ratio écran / ClavPad qui permettrait de moins bouger le poignet.
- Ensuite, effectuer un contrôle de la souris comme le touchpad et non pas un contrôle absolu.
- Et enfin, faire des tests sur la zone "tactile" du clavier (plus ou moins grande).