

Les écrans tactiles et Ubuntu

Quentin Bérard

Est-ce que Ubuntu supporte les écrans tactiles ? La réponse est oui, mais ... Le sujet est assez difficile, mais nous pourrions tout de même nous amuser un petit peu si vous possédez un écran tactile ! Si vous voulez utiliser toutes les capacités d'un écran tactile multi-touch, il faudra mettre les mains dans le cambouis.

Ce qu'il faut en retenir pour le moment, c'est le terme « en chantier » ou « en développement ». Les interactions ont changé depuis la précédente version de Unity et risquent de changer encore, de plus, sauf Unity, aucune application ne supporte le multi-touch par défaut. C'est pourquoi l'information est difficile à trouver sur ce sujet. Nous supposons que les développeurs ne veulent pas donner des spécifications susceptibles de changer rapidement ; nous n'entrerons donc, nous non plus, pas trop dans les détails. Nous nous trouvons principalement face à 2 questions :

- Est-ce que les écrans sont supportés matériellement ? : disposons-nous des pilotes pour communiquer avec eux ? La réponse est « généralement oui », le support a grandement augmenté avec Maverick puis avec Natty. Pour savoir si un modèle en particulier est compatible, rendez-vous sur <https://wiki.ubuntu.com/Multitouch/HardwareSupport> Les bricoleurs savent probablement que des écrans tactiles multi-touch (capacitifs) sont en vente sur e-bay. Ils peuvent reconnaître 2 doigts, sont en taille 9 ou 10 pouces et sont compatibles pour leur majorité avec Ubuntu.
- Quelles sont les interactions possibles ? Si vous avez un écran non multi-touch (qui peut reconnaître un seul doigt), la réponse est assez simple : vous pouvez cliquer ou utiliser des applications multi-touch (voir ci-dessous). Si vous avez un écran multi-touch, cela devient un peu plus compliqué et cet article vous concerne.

Les interactions tactiles avec l'écran

La problématique de l'équipe de développement est de définir une IHM : interaction homme-machine, qui soit intuitive et simple.

Il y a principalement deux types d'interactions. Celles gérées par Ubuntu : manipulations des fenêtres et de Unity, et celles concernant les applications : faire défiler un texte, bouger une photo, faire un clic droit, une rotation, un zoom, ...

Les interactions au niveau Bureau

Les interactions gérées par Ubuntu suivent une certaine logique : Canonical a décidé de séparer les comportements selon le nombre de doigts qui interviennent :

- 1 doigt : un geste pour un objectif précis,
- 2 doigts : mouvement au niveau de l'application,
- 3 doigts : mouvement au niveau de l'écran,
- 4 doigts : mouvement au niveau de Unity.

Voici les interactions disponibles :

- Déplacer une fenêtre : poser 3 doigts, attendre un instant et déplacer la fenêtre n'importe où.
- Démaximiser : soit poser 3 doigts et les rapprocher, soit déplacer la fenêtre.

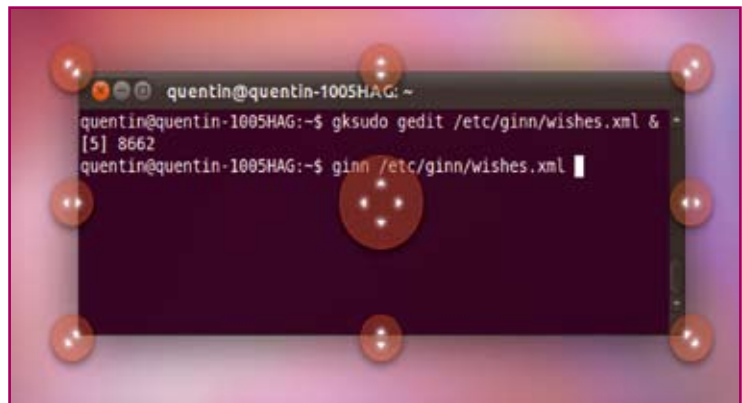


Figure 1. Unity Multi-Touch Grab Handles en action !



Figure 2. Les gestes par défaut de All In One Gesture (greffon pour Firefox)

- Agrandir une fenêtre : poser 3 doigts et les écarter ou la déplacer vers le haut.
- Faire occuper la moitié droite / gauche de l'écran à une application : déplacer la fenêtre vers la droite / la gauche.
- Redimensionner une fenêtre : une tape à 3 doigts, cela affiche les « grab handles ».
- Afficher le dash : une tape à 4 doigts.
- Afficher / masquer le launcher : faire glisser 4 doigts vers la droite / gauche.

Les interactions au niveau application

Elles concernent les interactions gérées par les applications, nous avons encore 2 catégories :

- Soit l'application supporte nativement les écrans tactiles et le multi-touch.
- Soit l'application ne le supporte pas, mais nous pouvons alors forcer une « reconnaissance » de certains mouvements en attendant qu'elle soit pleinement compatible. C'est sur ce point là

que portera la plus grande partie de cet article.

Interactions supportées nativement Evince, EOG

Pour l'instant, aucune des applications installées par défaut ne supporte nativement les interactions avancées avec les écrans tactiles. À des fins de test, deux applications ont néanmoins été rendues compatibles, il s'agit de Evince, un logiciel qui permet de visualiser les documents pdf et ps ; et l'application EOG (Eye of GNOME) qui permet de visualiser les images.

Ces versions spéciales permettent de déplacer les documents et photos de manière fluide avec un seul doigt, le zoom et la rotation sont aussi possibles.

Cependant attention : il se peut que les interactions soit aussi actives sur un pavé tactile, dans ce cas il n'est plus possible de sortir la souris du document, tout mouvement sur le pavé tactile fait uniquement bouger le document ! (pour s'en sortir, faire un clic droit qui permet à nouveau au curseur de bouger). Nous observons donc que ces applications sont plutôt destinées aux tablettes tactiles (qui n'ont donc pas de pavé tactile !), ou aux PC équipés de souris.

Firefox

Deux extensions peuvent servir au support de l'écran tactile pour Firefox, mais il ne faut pas utiliser les deux simultanément !

- Soit utiliser « Grab and Drag ». Cette extension permet de naviguer à un doigt dans les pages web. Il y a également un effet d'inertie qui permet à la page de continuer de défiler quelques secondes après avoir donné une « impulsion ». Pour l'installer, rendez-vous sur cette page : <https://addons.mozilla.org/fr/firefox/>

addon/grab-and-drag/ Lors de l'installation, une question vous est posée : « l'm using Grab and Drag with a ... », choisir « pen-based computer (...) ». Ensuite, allez dans les préférences de Grab And Drag. Les réglages par défaut du « Momentum » (option qui permet de continuer à faire défiler le texte après avoir retiré le doigt) ne fonctionnent pas très bien pour mon écran tactile, nous vous proposons de mettre les 2 premiers curseurs au minimum, et le 3^{ème} à un quart. Vous pourrez ensuite configurer ginn pour implémenter « page suivante » et « page précédente » à l'aide de mouvements à 2 doigts.

- Soit utiliser All in One gesture disponible ici : <https://addons.mozilla.org/fr/firefox/addon/all-in-one-gestures/> Cela permet des actions avancées avec un doigt. Prenez quand même le temps de faire un petit réglage : par défaut, les actions sont à effectuer avec un clic droit. Pour utiliser le clic gauche (le touché de l'écran) à la place, allez dans les préférences de l'extension, « Préférences avancées #1 », puis sélectionnez « gauche » en haut de la fenêtre. Vous pourrez ensuite configurer ginn pour vous déplacer à l'intérieur de la page à l'aide de 2 doigts.

Alors, quelle extension choisir ? Cela dépend de vos préférences. Alors que la première permet un déplacement aisé dans la page, la seconde permet d'avoir plein de raccourcis à portée de doigt. Essayez les deux et prenez ce qui vous convient le mieux !

Interactions injectées dans les applications

Vient maintenant l'autre catégorie d'applications, celle où nous injectons des mouvements. Voici comment cela se passe : l'application ginn est à l'écoute de l'écran pour toutes les applications qui ne

supportent pas les écrans tactiles nativement. Elle reconnaît les mouvements réalisés par l'utilisateur puis transmet certaines instructions à l'application : des instructions bien connues par celle-ci comme un raccourci clavier, ou un clic de souris. Cela permet de rendre absolument toutes les applications compatibles mais il y a deux inconvénients :

- les interactions ne sont pas forcément aussi fluides que si l'application gérait nativement le tactile,
- et il faut personnaliser manuellement l'application pour les interactions avancées de chaque application.

Ginn contient un fichier de configuration par défaut sous Ubuntu, mais je le trouve pas très bien adapté, nous allons donc vous expliquer comment le modifier et le personnaliser, vous verrez que les possibilités sont déjà assez étendues.

Personnaliser ginn

Présentation

Pour personnaliser ginn, il n'existe pas encore d'utilitaire graphique, uniquement un fichier de configuration. Une référence est présente ici : `/usr/share/ginn/wishes.xml`. Mais le fichier que nous modifierons sera `/etc/ginn/wishes.xml`.

Ginn peut reconnaître différents types de mouvements :

- déplacement à 1, 2, 3 ou 4 doigts,
- le pincement à 2, 3 ou 4 doigts,
- la rotation à 2 doigts,
- et la simple tape (un touché) à 1, 2, 3 ou 4 doigts.

Et nous pouvons donc assigner soit un clic, soit un raccourci clavier à chacun de ces mouvements. Le fichier nous permet donc de faire les correspondances mouvement - effet pour la globalité des applications ; pour obtenir un comportement plus spécifique à certaines applications, il conviendra également de l'inscrire dans ce fichier.

Édition du fichier de configuration

Comme nous pouvons le voir dans `/usr/share/ginn/wishes.xml`, une configuration est proposée, nous ne la trouvons pas adaptée, mais cela donne une base pour observer la façon dont est réalisé ce fichier.

Alors, commençons par la structure du fichier, la base est comme le représente Script 1. Il y a donc une partie globale et une partie spécifique à chaque application.

Nous n'avons maintenant plus qu'à apprendre comment remplir les parties `<!--` Assignation mouvement `<-->` effet `-->`.

Chaque assignation d'un mouvement a une action à l'apparence disponible dans le Script 2.

Détaillons les balises ainsi que leurs attributs :

- Balise *wish* :
 - Attribut *gesture*, il faut choisir parmi cette liste : *Drag* (glissement), *Pinch* (pincement), *Rotate*, *Tap*. Il s'agit du mouvement considéré, vous l'aurez compris !
 - Attribut *fingers* correspondant au nombre de doigts entrant en jeu, entre 1 et 4.
- Balise *action* :
 - Attribut *name*, mettez-y le nom que vous voulez.
 - Attribut *when* : est-ce que l'effet doit se produire une seule fois à la fin du mouvement : *finish* ? Ou doit-il être émis plusieurs fois : *update* ? La première option sert typiquement à afficher tou-

tes les fenêtres, alors que la seconde sert à faire défiler une page par exemple.

- Balise *trigger* : nous avons ici des propriétés qui dépendent du « gesture » adopté, nous les verrons ci-dessous.
- Balise *key* : cette balise décrit l'action effectuée lorsque le mouvement est reconnu. Entre les deux balises *key* se trouve la touche, et nous pouvons éventuellement ajouter un modificateur comme *Control* en attribut « modifier1 », voire 4 modificateurs : *modifier1*, *modifier2*, *modifier3*, *modifier4*. Les modificateurs, quant à eux; peuvent être : *Control_L*, *Control_R*, *Shift_L*, *Shift_R*, *Alt_L*, *Alt_R*, *Super_L*, *Super_R*. [*Shift*] étant la touche *Majuscule* et [*Super*] la touche qui comporte généralement un Logo avec une fenêtre ! Vous aurez compris que `<key modifier1="Super_L">w</key>` lance "super-w", (c'est un raccourci qui permet d'afficher toutes les fenêtres) !
- Balise *button* qui doit s'écrire à la place de la balise *key*. Elle permet d'injecter un bouton de la souris, par exemple : 1 (clic gauche), 2 (clic molette), 3 (clic droit), 4 (roulette vers le haut), 5 (roulette vers le bas), 6 (roulette vers la gauche), 7 (roulette vers la droite), 8 (précédent dans un navigateur) et 9 (suivant dans un navigateur). Nous pouvons également appliquer les « modifier » vus précédemment à cette balise. Les

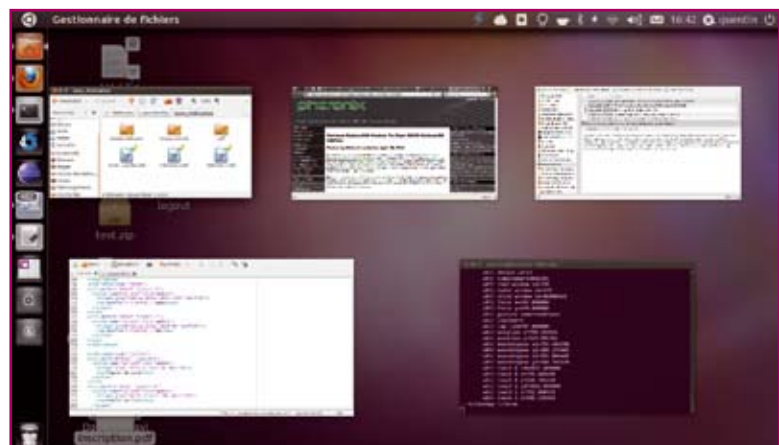


Figure 3. Exposé, *super-w*, et un raccourci tactile ?

Script 1. Structure globale du fichier wishes.xml

```
<ginn>
  <global>
  <!-- Assignation mouvement <-> effet -->
  </global>
  <applications>
    <application name="inkscape">
  <!-- Assignation mouvement <-> effet -->
  </application>
    <application name="une autre application">
  <!-- Assignation mouvement <-> effet -->
  </application>
  </applications>
</ginn>
```

Script 2. Structure de la balise wish : assignation mouvement <-> effet

```
<wish gesture="Drag" fingers="2">
  <action name="action5" when="update">
  <trigger prop="delta y" min="20" max="80"/>
  <key modifier1="Control_L">Down</key>
  </action>
</wish>
```

événements roulette sont particulièrement utiles !

Il ne nous reste plus que deux détails à régler. Les touches qui, à l'intérieur d'une balise key, peuvent être :

- les lettres de a à z et de A à Z,
- un chiffre (au dessus du clavier) : 0..9,
- une touche du pavé numérique : KP_0 à KP_9, KP_Add, KP_Subtract, ...
- une flèche : Up, Right, Left, Down,
- les touches spéciales : Back-Space, Tab, Return, Escape, Delete, Page_Up, Page_Down, End, Begin,
- un caractère spécial : space, plus, minus,
- une touche décrite dans le fichier `/usr/include/X11/keysymdef.h` à laquelle a été enlevé le « XK_ » qui précède chaque touche. Pour avoir ce fichier, installez `x11proto-core-dev`.

Les propriétés que peut prendre la balise trigger sont, suivant le mouvement reconnu (nous indiquons des valeurs en exemple entre parenthèses) :

- Drag : delta x (-9), delta y (22), velocity x (-0.64), velocity y (1.6), position x (634), position y (482),
- Rotate : angle delta (-0.36), angular velocity (-0.03), angle (-1.16),
- Pinch : radius delta (-45), radial velocity (-3.2), radius (180.5),
- Tap : tap time (74), position x (741), position y (469).

Le mieux est de s'inspirer des exemples donnés et si les mouvements ne vous conviennent pas, essayez de toucher à ces réglages.

Voilà, vous savez configurer ginn, je vous donne un premier exercice pratique : pour l'instant 2 doigts qui glissent sur l'écran émettent des événements « flèche », vous pouvez maintenant émettre des événements roulette, ce qui correspond plus au comportement attendu. Vous pouvez aussi décider que dans toutes les applications un pincement émet les touches [Ctrl-plus] ou [Ctrl-moins]. C'est un raccourci assez fréquent pour le zoom et le dézoom ! Nous pouvons aussi imaginer que poser 3 doigts pour les rapprocher affiche toutes les fe-

nêtres (« Exposé »), comme c'était le cas pour Maverick.

Enfin, pour lancer ginn, tapez `ginn /etc/ginn/wishes.xml` dans une console, ou ajoutez-le dans la liste des programmes lancés au démarrage.

Conclusion

Comme vous l'avez remarqué, les pistes sont bonnes, mais tout n'est pas encore bien finalisé. Un utilitaire graphique devrait apparaître pour configurer plus facilement ginn, il sera bienvenu. Mais surtout, de plus en plus d'applications seront nativement compatibles avec les écrans multi-touch !

Pour conclure cet article, nous vous donnons encore un petit bonus et des pistes pour aller plus loin.

Manipulation des fenêtres

Après tant de choses compliquées, voici une chose simple et bien pratique !

Nous avons aperçu au début de cet article une prise d'écran de « Unity MT Grab Handles ». Le rond central permet de déplacer la fenêtre, tandis que les 8 autres permettent de changer sa taille. Très pratique pour les écrans tactiles ! Comme indiqué, un raccourci est disponible par défaut, mais il nécessite un écran qui puisse reconnaître 3 doigts. Si vous voulez plutôt un raccourci clavier, installez `compizconfig-settings-manager` (le gestionnaire des propriétés de Compiz). Ensuite lancez-le (allez dans les *Réglages du Système*, ou tapez `ccsm` dans le dash). Nous voyons en bas l'option « Unity MT Grab Handles » qui nous permet d'y affecter un raccourci clavier.

Aller plus loin

Si vous êtes curieux et voulez en savoir plus, nous vous conseillons ces pistes (en anglais) :

- Wiki anglophone : <https://wiki.ubuntu.com/Multitouch>
- Ligne directrice des mouvements Unity : https://docs.google.com/View?id=dfkkjccj_1482g457bcc7
- Site de développement de ginn : <https://launchpad.net/ginn> ■