

## 8: Saisie au clavier et à la souris

### *Tapez, cliquez et faites glisser*

Dans ce didacticiel, nous allons examiner comment contrôler des actions Max à l'aide de deux contrôleurs courants: la souris et le clavier de votre ordinateur. Nous verrons également comment convertir des nombres en lettres et identifier des frappes au clavier particulières.

Il n'est pas toujours pratique - ni même utile - d'utiliser du matériel externe pour contrôler les patchs Max. Parfois, l'ordinateur lui-même contient tous les mécanismes de contrôle dont vous avez besoin pour utiliser vos patchs. Travailler avec la souris et le clavier peut nous donner un contrôle étendu de nos patchs sans avoir à vous soucier de tout autre dispositif.

Pour ouvrir le patch du didacticiel, cliquez sur le bouton vert **Ouvrir didacticiel** dans le coin supérieur droit de la fenêtre de documentation.

### *Regarder la souris*

Jetez un coup d'œil au patcheur du tutoriel. Sur le côté gauche de la fenêtre se trouve un petit patch qui relie un objet *metro* à deux nouveaux objets Max nommés *mousestate* et *modifiers*. Cliquez sur le *toggle* pour allumer le *metro*, puis déplacez votre souris sur l'écran. Vous remarquerez que les boîtes de *nombres* connectées à l'objet *mousestate* rapportent les informations lors du déplacement de la souris et que le *toggle* connecté est activé chaque fois que vous cliquez sur le bouton de la souris.

La sortie de l'objet *mousestate* fournit des informations sur l'état du bouton et la position du curseur de la souris/curseur lorsque le curseur est positionné dans une fenêtre du Max Patcher. Il fournit une valeur indiquant si le bouton de la souris est "cliqué", ainsi que deux ensembles de coordonnées: la position absolue de la souris sur les axes x et y et une position delta montrant le mouvement depuis la dernière fois que *mousestate* a été interrogé. La position absolue est basée sur les coordonnées de l'écran, et non sur la fenêtre du patcheur. Vous pouvez ainsi suivre le mouvement n'importe où sur l'écran de votre ordinateur. Cela signifie également que les résultats de *mousestate* varieront en fonction de la taille de votre écran d'ordinateur. Les coordonnées de l'écran sur l'ordinateur sont des cartésiennes inversées, c'est-à-dire que les coordonnées supérieures gauche de votre écran seront **0, 0** et que les coordonnées inférieure droite correspondront à la taille de votre écran moins 1 (par exemple, un affichage 1024x768, vous donnera les coordonnées **1023, 767** lorsque la souris se trouve sur le bord inférieur droit).

La valeur delta indique les changements de position de la souris - mais quand les changements sont-ils réellement signalés ? Les modifications (ainsi que la position absolue) ne sont signalées que lorsqu'un message **bang** est reçu dans l'entrée de l'objet, d'où l'utilisation de l'objet *metro* pour générer des messages. Cette opération s'appelle le **polling**. Elle vous permet de contrôler la densité des messages créés par un objet. Ceci est particulièrement important dans le cas d'un objet tel que *mousestate*, où des flux massifs de données indésirables pourraient être générés s'il agissait automatiquement. La seule sortie qui ne nécessite pas de polling actif est la notification du bouton de la souris (sortie de gauche de l'objet).

### *Observer le clavier*

L'objet *modifiers* est adjacent à l'objet *mousestate*. Cela permet de suivre (de gauche à droite) les touches shift et caps-lock de votre clavier, ainsi que les touches Cmd, Ctrl et Option sur un système Macintosh ou les touches Alt, les clicks droits de la souris ou la touche Ctrl sur un Système

Windows. Toutes ces sorties nécessitent également une interrogation, mais elles constituent un excellent moyen de capturer l'état des touches de « modificateurs » en tant que cas particuliers de votre patch.

Les trois petits patches suivants permettent de capturer les pressions sur les touches du clavier de votre ordinateur. Contrairement aux messages de la souris, les frappes au clavier produisent une sortie uniquement lorsqu'elles sont enfoncées et relâchées. Il n'est donc pas nécessaire d'utiliser l'interrogation pour réduire la densité des messages. Dans le patch le plus à gauche, l'objet *key* est utilisé pour obtenir la valeur ASCII d'une touche lorsqu'elle est enfoncée. Cette sortie est envoyée à une boîte de *nombres* (pour afficher la valeur), puis transmise à un objet *itoa* - qui convertit cette valeur entière en un symbole alphanumérique. Celui-ci est envoyé à la Console Max (via un objet *print*), où vous pouvez voir la lettre de la touche pressée. Essayez de taper votre nom et de regarder la Console Max; remarquez que les entiers sortant de l'objet *key* sont correctement convertis en leur symbole correspondant. Les touches générant des caractères **non imprimables**, p. ex. la touche de retour (ASCII **13**) et la touche de tabulation (ASCII **9**) génèrent les valeurs numériques correctes à partir de l'objet *key*, mais impriment une seule ligne vierge dans Console Max.

Juste à droite se trouve un autre petit patch qui utilise l'objet *keyup*. C'est presque identique à l'objet *key*, sauf qu'il ne rapporte aucune information sur la touche tant que la touche n'a pas été relâchée. Ceci est utile dans les situations où vous souhaitez qu'une action ne se produise que lorsque vous appuyez sur une touche. vous pouvez activer l'action lorsque vous recevez un message de *key*, puis le désactiver lorsque vous recevez un message de *keyup*.

Le petit patch suivant alimente la sortie d'un objet *key* en un objet appelé *numkey*. C'est un objet très pratique qui vous permet de travailler avec des entrées numériques au clavier dans le domaine des nombres entiers. Commencez par taper quelques chiffres - vous verrez qu'ils sont accumulés dans un nombre de la sortie de droite. Si vous appuyez sur la touche retour, vous verrez que le nombre est rapporté comme nombre entier à partir de la sortie de gauche. Cela signifie que l'objet *numkey* exécute toute l'accumulation et la traduction des valeurs du clavier dans une sortie numérique. Notez que ce patch est limité à la sortie d'un nombre entier. Si vous voulez une sortie en virgule flottante, vous devez ajouter un argument en virgule flottante (comme 0.0) à l'objet *numkey* pour lui indiquer de respecter la virgule et de rapporter une sortie en virgule flottante.

### *Contrôle de la sortie avec mousefilter*

Enfin, le patch le plus à droite montre un objet appelé *mousefilter* en action. Lorsque vous cliquez et faites glisser le *curseur* supérieur, nous constatons qu'aucune sortie n'est générée tant que le bouton de la souris n'est pas relâché. Il s'agit d'un autre objet pratique qui peut être utilisé pour limiter la génération de données intermédiaires indésirables. Dans ce cas, il se peut que vous ne souhaitiez pas effectuer d'action tant que le mouvement du *curseur* n'est pas terminé. En faisant passer la sortie de l'objet *slider* via *mousefilter*, toute sortie est supprimée jusqu'à ce que le bouton de la souris soit relâché.

### *Création d'un patch "intelligent"*

Utilisons certains de ces objets et techniques pour créer un patch qui fonctionne "intelligemment" en fonction de l'activité du clavier et de la souris. Créez deux objets *metro* avec des arguments de **250**, et connectez les objets *boutons* à leurs sorties - ce seront nos sujets de test. En utilisant le patch le plus à gauche comme point de départ, connectez la sortie la plus à gauche de *mousestate* (qui indique l'état du bouton de la souris) à l'entrée gauche des deux objets *metro*. Verrouillez le patch et vérifiez que le fait d'appuyer sur le bouton de la souris allume les objets *metro* et que leur relâchement provoque l'arrêt des objets *metro*.

Suivons ensuite la position horizontale de la souris à l'aide d'un *curseur*. Ajoutez un curseur au patch et connectez-le à la deuxième sortie du *mousestate*. Pour que cela ait un sens, nous devons changer la **plage** du curseur. Utilisez le message de taille (ou utilisez l'inspecteur d'objets) pour modifier la portée maximale du curseur afin qu'elle corresponde à la largeur de votre écran en pixels. Ce sera quelque chose comme **800**, **1024** ou **1280** pixels (en fonction de la résolution de votre écran). Maintenant, si vous activez l'objet *metro* qui interroge le *mousestate*, vous devriez voir le curseur bouger en même temps que les mouvements horizontaux de votre souris.

Utilisons cette valeur pour contrôler la vitesse de sortie des deux objets *metro*. Connectez la sortie du curseur à l'entrée droite du premier *metro* que vous avez créé. Maintenant, lorsque vous déplacez la souris (en maintenant le bouton enfoncé), vous verrez la vitesse du premier *metro* changer en fonction de la position actuelle de la souris. Utilisons également le curseur pour contrôler la vitesse du deuxième objet *metro*, mais mis à l'échelle à l'aide d'un opérateur mathématique. Ajoutez un objet *\** au patch avec un argument de **2** et connectez-le entre le curseur et l'entrée droite du second objet *metro*. Maintenant lorsque vous déplacez la souris (et appuyez sur le bouton de la souris), vous verrez que le deuxième *metro* met deux fois plus de temps à se déclencher que le premier, mais les deux sont contrôlés par la position de la souris.

Enfin, utilisons le clavier pour changer l'échelle de cet objet *\**. En utilisant la sortie de la sortie gauche de l'objet *numkey* (celle qui ne signale les valeurs qu'après avoir appuyé sur la touche retour), nous pouvons le connecter à l'entrée droite de l'objet *\**. Maintenant, nous pouvons changer l'échelle du deuxième *metro* en entrant un nombre à l'aide des touches numériques – et ce sans avoir à cliquer sur la boîte de *nombres* pour effectuer la modification. Vous pouvez tester cela en saisissant un nombre (essayez **3** par exemple), en appuyant sur la touche retour, puis en testant la fonction de clic de souris. Vous devriez voir les objets *metro* réagir à la position actuelle de la souris.

## *Résumé*

Ce tutoriel montre comment l'ordinateur lui-même peut être utilisé comme contrôleur pour votre patch Max. Plutôt que de dépendre uniquement des éléments de l'interface utilisateur, vous pouvez interroger directement la souris et le clavier et utiliser les valeurs résultantes pour modifier les actions de votre travail.