

51-Instruments

Hébergement d'instruments avec *vst* ~

L'objet *vst* ~ peut également héberger vos synthétiseurs plug-ins préférés. Dans le tutoriel précédent, nous avons vu comment charger des plug-ins d'effets et accéder à leurs différentes fonctionnalités. Pour résumer, il existe quatre façons de charger un plug-in:

.Cliquez sur l'icône de plug-ins dans la barre d'outils de gauche pour ouvrir le navigateur de plugs. Il présente une liste de tous les plug-ins avec des options pour affiner la liste par type et une fonction de recherche complète. Notez que cela peut être fait avec le patcheur verrouillé ou déverrouillé.

.En mode d'édition de chemin, cliquez sur le menu d'action d'objet situé à gauche de l'objet *vst* ~, maintenez le pointeur de la souris sur l'option de fichiers et sélectionnez le plug-in désigné dans la liste qui apparaît.

.Envoyez à l'objet *vst* ~ un message **plug**. Celui-ci peut prendre le nom d'un plug-in comme argument. Si aucun n'est fourni, un dialogue d'ouverture de fichier apparaîtra.

.Ajoutez le nom du plug-in comme argument à l'objet *vst* ~.



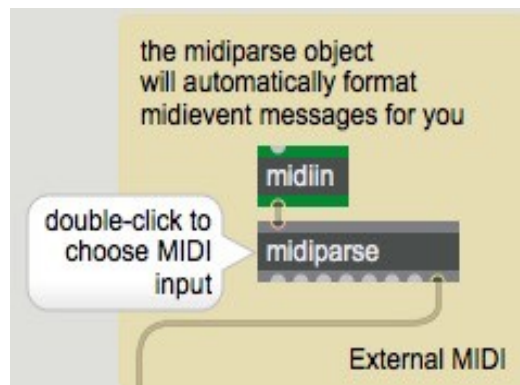
Un plug typique de synthé virtuel

Une fois chargé, les fonctionnalités d'un plug-in d'instrument sont accessibles de la même manière que les plug-ins d'effets. Le message **open** ouvre l'interface de l'instrument (l'option générique est disponible), et les listes de paramètres et de programmes sont disponibles comme auparavant. La seule différence est que les instruments ne traitent pas normalement l'audio. Ce qui est requis, ce sont des commandes pour jouer des notes. Les instruments plug-in ayant été conçus à l'origine pour être utilisés dans des séquenceurs MIDI, il n'est pas surprenant que des messages MIDI soient utilisés pour les jouer.

Pour envoyer un message MIDI à un instrument dans *vst* ~, nous l'incluons comme argument d'un message **midievent** (par exemple, si nous envoyons le message **midievent 144 60 127**, l'instrument devrait commencer à jouer la note 60 (le C du milieu) puissamment. Cette note doit continuer à être jouée jusqu'à ce que nous envoyions le message **midievent 128 60 64** ou **midievent 144 60 0**). Max simplifie la tâche consistant à envoyer des messages **midievent** correctement formatés à un instrument dans *vst* ~ en utilisant les objets *midiparse* et *midiformat*.

Entrée MIDI et instruments VST

L'entrée MIDI brute peut être automatiquement formatée en messages **midievent** en utilisant l'objet *midiparse*. L'objet reçoit l'entrée MIDI brute d'un objet *midiiin* et sort les messages **midievent** appropriés sous forme de liste à partir de sa sortie la plus à droite. Connectez cette sortie à l'entrée gauche de l'objet *vst* ~ et vous êtes prêt.



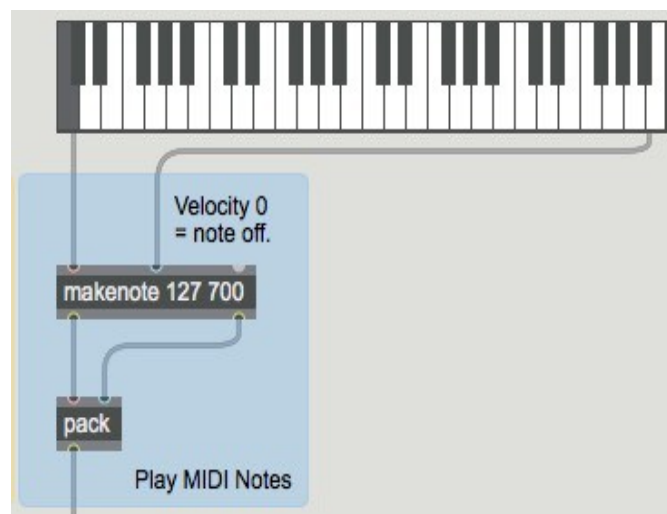
Entrée MIDI externe

Messages MIDI et instruments VST

Vous n'utilisez pas toujours l'entrée MIDI de sources externes pour envoyer des messages à un instrument dans un objet *vst* ~. L'objet *midiformat* a également la capacité de formater et de sortir des messages **midievent** qui peuvent être envoyés à l'entrée gauche de l'objet *vst* ~.

- Ouvrez le patch du didacticiel pour voir quelques méthodes de génération de **midievents**.

Lecture de notes MIDI



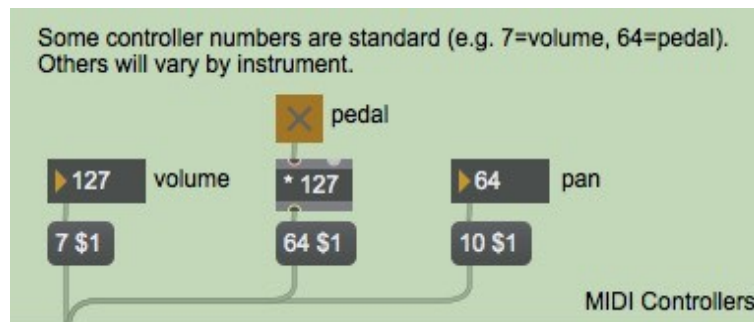
Formatage des **midievents** pour jouer des notes

La section bleue du patch du didacticiel présente une stratégie pour générer des notes. Fondamentalement, tout ce qui est créé par *makenote* est facilement géré par un objet *pack* et un objet *prepend*. L'objet *pack* crée une liste composée du numéro de note MIDI et d'une valeur de vélocité, qui est envoyée à l'entrée la plus à gauche de l'objet *midiformat*. Cette méthode fonctionnera avec le comportement de *makenote* qui envoie une vélocité de 0 pour la note off. Tout autre système de génération de note que l'on souhaite concevoir fonctionnera également, tant que le résultat est une liste de numéro de note et de vélocité.

Contrôleurs MIDI

La section verte du patch de didacticiel montre un exemple de messages de changement de contrôle. Les messages de changement de contrôle se présentent sous la forme d'une liste dont la première

valeur identifie le numéro du contrôleur, et dont la deuxième valeur représente la valeur du contrôleur. Certains numéros de contrôleur MIDI sont standard pour tous les instruments VST (comme le contrôleur 64, qui est une pédale de sustain, ou le contrôleur 7, qui est la commande de volume), tandis que d'autres varient d'un instrument à l'autre.



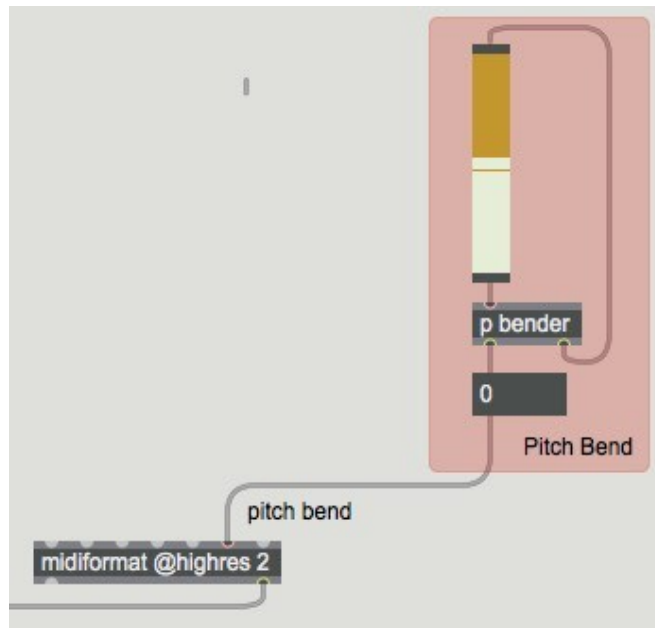
Mise en forme des messages de changement de contrôle

La spécification MIDI stipule que les contrôles de type interrupteur, comme les pédales, doivent être désactivées avec une valeur de 0 et activées avec 127. Certains instruments de plug-in interpréteront toute valeur non nulle comme étant activée, mais vous ne pouvez pas compter dessus, donc nous multiplions la sortie du *toggle* par 127 pour envoyer le message spécifié. En outre, un ensemble de numéros de contrôle qui est défini de manière rigide est le dernier 8, compris entre 120 et 127. Il s'agit des messages de **mode de canal** qui affectent l'instrument dans son ensemble. Les plus utiles sont 120: **tous les sons désactivés**, 121: **réinitialisation des contrôleurs** et 123: **toutes les notes désactivées**. Les autres, qui définissent des caractéristiques comme la polyphonie et le contrôle local, ne s'appliquent pas vraiment aux plug-ins. Ces trois messages sont envoyés avec une valeur de 0.

Pitch Bend MIDI

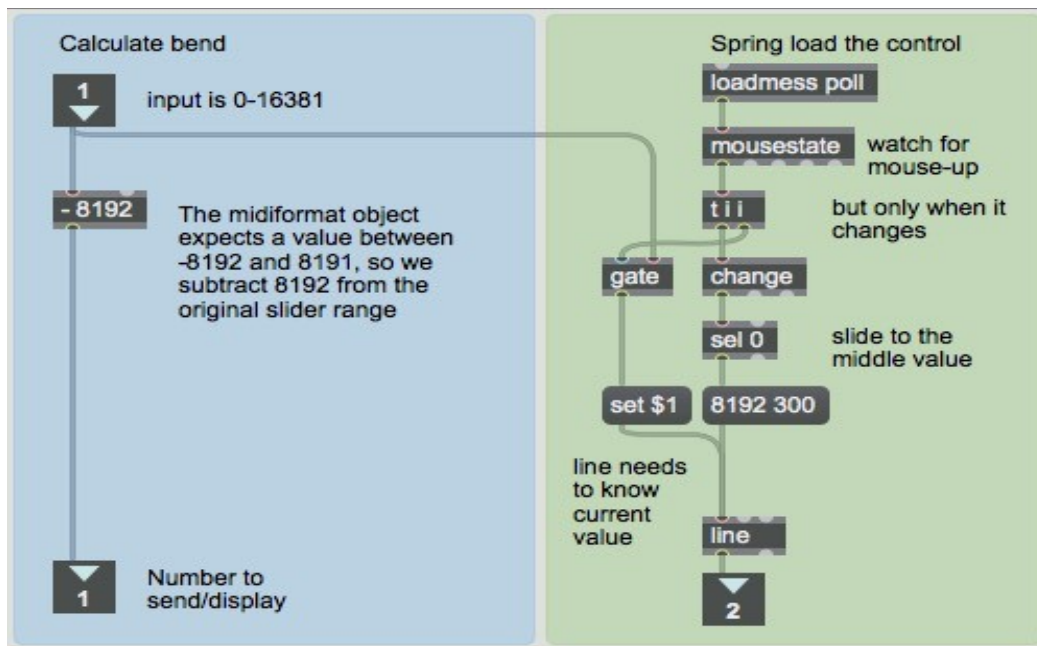
La section rose du patch de didacticiel montre un exemple d'envoi de messages de pitchbend haute résolution (14 bits). La technique implique à ajouter l'attribut **hires** comme qu'argument au *midiformat* (c'est-à-dire @ **hires 2**), et à utiliser le sub-patcher **p bender** pour envoyer une valeur à l'entrée pitch bend de l'objet *midiformat*.

Dans cet exemple, nous définissons la valeur de l'attribut @**hires** à une valeur de **2**, ce qui correspond à une résolution de pitch bend MIDI traditionnelle de 14 bits. La section rose du patch du didacticiel en montre un exemple. En plus de définir la valeur de l'attribut comme argument de l'objet *midiformat*, nous utilisons le sub-patch **p bender** pour envoyer une valeur comprise dans la plage de -8192 à 8191 (c'est-à-dire 16384 valeurs possibles) à l'entrée pitch bend de l'objet *midiformat*.



Utilisation de midiformat pour envoyer des pitch bends

Le contenu du patcheur **p bender** montre une façon de procéder. L'objet *slider* du patch du didacticiel a reçu une taille de 16384 pas (c'est une résolution de 14 bits, et c'est typique de la plupart des synthétiseurs). La plage que nous envoyons inclut la notion d'une position "centrale" pour aucun pitch bend (0). Le sub-patch **p bender** contient un bit supplémentaire de patching qui ramène le contrôle pitchbend à 0 lorsqu'il est relâché (sinon, tout bend est laissé en place pour toujours). Le sub-patch s'en charge pour nous en détectant un mouvement de souris vers le haut et en déclenchant un objet *line* pour ramener en douceur le contrôle à son centre. La disposition des messages *trigger*, de *gate* et *set* est nécessaire pour que l'objet *line* soit informé de la position actuelle du *slider* afin que la rampe commence à cet endroit.



Logique de contrôle Pitchbend

Remarque: une autre méthode pour envoyer des informations de pitchbend serait de définir l'attribut **@hires** sur une valeur de **1**. Lorsqu'il est défini, l'objet midiformat attendra une

*entrée sous la forme d'une valeur à virgule flottante comprise entre -1,0 et 1,0. Vous trouverez peut-être que c'est une façon plus intuitive de travailler (et vous devrez modifier le patch de **p bender** en conséquence).*

Résumé

Max peut accueillir des synthétiseurs logiciels aux formats VST ou AudioUnit avec l'objet `vst ~`. Toutes les fonctionnalités de l'un ou l'autre format sont disponibles pour le programmeur Max.