

31-Synthétiseur MIDI

Dans ce tutoriel, nous montrons comment créer et utiliser un simple synthétiseur à 4 voix contrôlable par MIDI. En cours de route, nous aborderons des méthodes simples de routage polyphonique ainsi que l'utilisation d'une variété de messages MIDI communs pour contrôler les paramètres de synthèse.

- Pour utiliser les patches de didacticiel de cette section du didacticiel, assurez-vous qu'un contrôleur MIDI correctement configuré est connecté à votre ordinateur. Les didacticiels de cette section utilisent une variété de messages MIDI comme exemple d'entrée; si votre contrôleur ne possède aucune de ces caractéristiques, vous pouvez simuler leur entrée avec les objets de l'interface utilisateur dans le didacticiel.

Jouer avec notre synthétiseur

- Jetez un coup d'œil au patcheur du tutoriel. Remarquez qu'il contient un peu de logique de patcheur impliquant quatre catégories différentes d'objets d'entrée MIDI qui envoient des messages et des signaux à quatre copies d'une abstraction appelée **synthvoice** ~. Activez l'audio en cliquant sur *l'ezdac* ~, montez le curseur de *gain* ~ et jouez quelques notes sur votre clavier MIDI. L'objet *kslider* devrait s'animer en réponse à vos actions. Tout en maintenant une note, ajustez la molette de pitchbend sur votre clavier. Ajustez la molette de modulation ou un autre contrôleur pouvant envoyer le CC#1. Écoutez les différents changements apportés au son.

Notre patcheur de tutorial traite le MIDI par le biais d'un certain nombre de méthodes. Examinons-les une par une.

- Jouez les notes sur votre clavier, une par une, et regardez la zone du patcheur du tutoriel étiquetée **1**. Remarquez ce que l'objet *poly* fait aux valeurs. Essayez de jouer une note et de la maintenir, puis ajoutez-en une autre pendant que la première touche est toujours enfoncée.

Notre patcheur de tutorial est capable de jouer quatre sons en même temps, car il y a quatre copies différentes de l'abstraction **synthvoice** ~ dans notre patch. Toutefois, afin de tirer profit de la polyphonie, nous devons trouver comment acheminer nos valeurs MIDI vers les différentes voix afin que la copie appropriée de l'abstraction de **synthvoice** reçoive chaque message. L'objet *poly* prend la hauteur et la vélocité MIDI et effectue l'assignation de la voix sur les valeurs basées sur les arguments de l'objet. La première note que l'objet reçoit se verra attribuer la voix numéro **1**, la seconde note la voix **2**, etc. Si l'on dépasse la polyphonie pour laquelle l'objet est programmé (dans notre cas, **4** voix), l'objet se retournera et recommencera à la voix **1**. Les événements note-off seront associés à la *même* voix que leurs événements note-on correspondants, garantissant que le message d'arrêt d'un événement MIDI va à la même destination que celui qui l'a démarré.

- En utilisant votre clavier MIDI, jouez un accord de quatre notes. Maintenant, sans relâcher aucune des touches, appuyez sur une cinquième note. Remarquez ce qui se passe.

Le deuxième argument de *poly* (**1**) lui dit de *voler* des voix si la polyphonie est dépassée. Si nous essayons de faire sonner plus de notes que ne le permet l'objet *poly*, la note la plus ancienne sera abandonnée et sa voix sera recyclée.

La sortie de notre objet *poly* est envoyée dans un objet *pack*, et ensuite à travers un objet *route* pour envoyer les listes restantes de hauteurs et de vélocités à la copie appropriée de notre abstraction **synthvoice**.

- Jetez un coup d'œil à la zone **2** du didacticiel. Déplacez la molette de pitch bend sur votre clavier contrôleur. Remarquez que, contrairement à la plupart des contrôleurs, son point d'arrêt se situe au milieu de la plage à **63**. Observez ce que fait la logique de patcheur ci-dessous fait pour mettre à l'échelle les valeurs comprises entre **-2.** et **2.**

Pour créer notre valeur de pitch bend, nous prenons la molette de pitch bend MIDI et nous divisons sa plage en deux objets d'échelle différents. L'objet de gauche prend la moitié inférieure de la plage MIDI et la met à l'échelle **-2.** à **0 .**; l'objet de droite met à l'échelle la plage supérieure entre **0** et **2**. Cela garantit que la valeur centrale dans la plage (**63**) correspond à une valeur de **0.** dans tous les cas.

- Jetez un coup d'œil à la zone de didacticiel intitulée **3**. Déplacez le contrôleur continu et regardez comment il est mis à l'échelle. Regardez la zone du didacticiel étiquetée **4**. Si vous disposez d'un contrôleur MIDI avec des capacités de transport en temps réel (c'est-à-dire qu'il peut envoyer une horloge MIDI), configurez-le pour qu'il transmette au tempo de votre choix et lancez le transport sur le contrôleur. Si vous le souhaitez, vous pouvez également utiliser un utilitaire de routage MIDI inter-applications et un séquenceur logiciel sur votre ordinateur pour effectuer cette opération. Sinon, cliquez dans la boîte de *nombre* intitulée **tick speed** et entrez la valeur **20**. Changez le contrôleur continu et jouez quelques notes. Écoutez le résultat. Changez la vitesse du **tick speed** en quelque chose de plus rapide (comme **10.**) Remarquez ce qui se passe.

Le CC#MIDI et les messages en temps réel sont envoyés par les objets *ctlin* et *rtin*, respectivement, pour travailler ensemble et contrôler un oscillateur basse fréquence (LFO). Les messages en temps réel qui définissent l'horloge rythmique contrôlent la **fréquence** du LFO; les messages du contrôleur en modifient la **profondeur**. Le MIDI beat clock fonctionne généralement à 96 PPQ; l'objet *timer* mesure les intervalles entre les ticks, qui sont ensuite mis à l'échelle pour obtenir le taux du LFO afin qu'il dure une mesure. Le message MIDI temps réel **250** (le message 'start' sur un séquenceur) réinitialise la phase de l'objet *cycle* ~ afin que le LFO se resynchronise avec une séquence si elle commence sur une barre de mesure.

Notez comment la sortie du LFO est mise à l'échelle de sorte que même si la profondeur envoie un signal de **0.**, le signal envoyé dans l'abstraction **synthvoice** ~ est garanti centré autour de **1**. Voyons ce que contient ce patch.

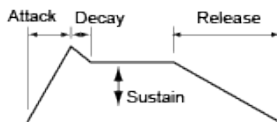
L'abstraction synthvoice ~

- Double-cliquez sur l'une des abstractions nommées **synthvoice** ~ et examinez la logique de patcheur à l'intérieur.

Notre abstraction **synthvoice** ~ possède trois entrées. La première entrée prend des listes de valeurs de hauteur et de vitesse de notre entrée clavier MIDI. La valeur de hauteur est envoyée pour piloter un signal constant (*sig* ~) auquel est ajouté le signal de l'entrée n° 2 (+ ~). Cette valeur est ensuite interprétée comme un nombre MIDI et convertie en fréquence dans le domaine du signal par un objet appelé *mtof* ~, qui se comporte comme l'objet *mtof* mais opère sur des signaux MSP continus au lieu de nombres Max. Cette valeur de fréquence alimente alors deux oscillateurs à bande limitée: une onde carrée (*rect* ~) et une onde en dents de scie (*saw* ~) qui sont mélangées ensemble. La fréquence de l'objet *rect* ~ est multipliée par le signal du LFO arrivant par l'entrée n° 3, ce qui permet d'obtenir un son riche et chorussé lorsque la profondeur du LFO est augmentée.

Enveloppes standard de synthétiseur: *adsr ~*

Pendant ce temps, la sortie de la vélocité des notes MIDI est mise à l'échelle entre **0.** et **1.** et envoyée à un objet appelé *adsr ~*. Cet objet signifie **Attack**, **Decay**, **Sustain**, **Release** et génère des rampes de signal dans une configuration standard empruntée à la conception des synthétiseurs analogiques:



Une courbe ADSR standard.

Les arguments de *adsr ~* sont interprétés comme un temps d'attaque, un temps de décroissance, un niveau de maintien et un temps de relâchement. Le niveau de maintien est un multiplicateur de l'amplitude globale que l'objet produit pendant une note de maintien.

L'objet *adsr ~* prend une valeur et l'interprète comme un déclencheur d'enveloppe d'une certaine amplitude. Tout nombre supérieur à **0** déclenche les phases d'attaque, de décroissance et de maintien, mis à l'échelle pour correspondre à l'amplitude du *trigger* (par exemple, une valeur d'entrée de **0,5** déclenche une enveloppe plus douce qu'un *trigger* de **0,8**). L'objet reste ensuite en phase de maintien, en émettant une valeur constante jusqu'à ce qu'il reçoive un **0**. Il passe ensuite vers la phase de relâchement de l'enveloppe et passe à **0.**

- Maintenant que nous comprenons comment les éléments du synthétiseur fonctionnent, revenez au patcheur principal et voyez si les paramètres contrôlables ont un sens. Essayez d'explorer toute la gamme des contrôles MIDI pour voir à quel point ils sont expressifs.

Résumé

Dans un patcheur, les événements de note MIDI peuvent être routés de manière polyphonique vers différentes copies de la même abstraction en utilisant l'objet *poly*. Des objets tels que *bendin* et *ctlin* peuvent être mis à l'échelle pour correspondre à différentes gammes de paramètres de synthétiseurs, et des commandes MIDI en temps réel peuvent être utilisées pour dériver des données de tempo pour les LFOs et les séquenceurs dans Max. L'objet *adsr ~* génère des rampes d'enveloppe basées sur des déclencheurs pour une note-on et une note-off, ce qui en fait l'outil idéal pour les systèmes basés sur les notes MIDI.