

32-Sampler MIDI

Dans ce tutoriel, nous verrons comment créer un échantillonneur contrôlable par MIDI dans MSP. En cours de route, nous examinerons de nombreux échantillonneurs, y compris le bouclage, les keymaps et les opérations multi-timbrales.

- Pour utiliser les patcheurs 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 pas l'une de ces fonctionnalités, vous pouvez simuler leur entrée avec des objets d'interface utilisateur du didacticiel.

Un échantillonneur amusant

- Jetez un coup d'œil au patcheur de ce tutoriel. Les deux ensembles de logique du patcheur (étiquetés **1** et **2**) traitent tous deux de la lecture d'échantillons stockés dans des objets *buffer* ~ en utilisant des objets *groove* ~ sous contrôle MIDI. Regardez la zone **1** du patcheur. Activez le son en cliquant sur *ezdac*~ et montez le curseur de *gain* ~. Dans l'objet *umenu* situé en haut du patch, sélectionnez l'option **1** ('bd+hh.aiff'). Jouez quelques notes sur votre clavier MIDI. Vous devriez entendre un échantillon de grosse caisse à différentes hauteurs selon la touche sur laquelle vous appuyez. Sélectionnez l'option **2** et écoutez, vous devriez entendre la caisse claire.

- Sélectionnez l'option **3** ('cym.aiff') dans le *umenu*. Jouez une note sur le clavier et maintenez-la enfoncée ou sélectionnez une note dans l'objet *kslider*. Remarquez que l'échantillon est en boucle. Sélectionnez l'option **4**; remarquez que le son de guitare basse met en boucle une houle. Jouez les sons déclenchés par les options **5** et **6**. Double-cliquez sur le fichier *coll* nommé **sampler**.

L'objet *umenu* de notre patcheur de didacticiel amène l'objet *coll* à déverser différentes données en fonction de l'échantillon sélectionné. En regardant son contenu en double-cliquant dessus, on peut voir l'ensemble de la base de données:

```
1,24 échantillon1 0 0 0; (bd + hh.aiff)
2, 33 échantillon2 0 0 0; (snare.aiff)
3, 50 échantillon3 0,136054 373,106537 1; (cym.aiff)
4, 67 échantillon4 60,204079 70,476189 1; (bass.aiff)
5, 84 échantillon5 0 0 0; (epno.aiff)
6,108 échantillon6 0 0 0; (ahkey.aiff)
```

Chaque entrée du fichier configure l'objet *groove* ~ du patcheur pour jouer un échantillon. Le format de chaque ligne du fichier *coll* est le suivant:

```
base_key buffer ~ _name loop_start loop_end looping
```

La **touche de base** fait référence à la note MIDI que l'échantillon jouera à une vitesse normale. Le deuxième élément dans *coll* fait référence à objet *buffer* ~ qui jouera l'échantillon. Les trois valeurs suivantes déterminent les points de début et de fin d'une boucle interne que l'objet *groove* ~ jouera, et s'il faut ou non l'utiliser (la dernière valeur).

Le but d'un échantillonneur est de lire des enregistrements (échantillons audio), souvent à partir d'un instrument acoustique. Si une source instrumentale acoustique est utilisée, il est inefficace de créer un échantillon unique pour chaque note possible de cet instrument. Au lieu de cela, quelques notes sont échantillonnées, et les hauteurs intermédiaires sont obtenues en jouant ces notes

échantillonnées légèrement rapidement ou lentement en fonction de leur note de base. De même, un échantillonneur est capable de jouer des notes qui durent bien plus longtemps que l'enregistrement d'un échantillon instrumental. Au lieu de cela, un échantillon de longueur modeste est utilisé, puisqu'une zone est trouvée **dans** la partie de maintien de l'échantillon qui peut être mise en boucle en toute sécurité. Lorsque vous jouez une note et la maintenez, l'échantillon commence la lecture au début; une fois qu'il se déplace dans la zone de boucle, il répète cette zone encore et encore; lorsque vous reprenez la note, elle est lue depuis la zone de boucle jusqu'à la fin de l'échantillon.

- Cliquez à nouveau sur *l'umenu* et remarquez comment les différents paramètres provenant de l'objet *coll* configurent l'objet *groove* ~.

La **touche de base** est convertie de MIDI en fréquence par l'objet *mtof* et utilisée comme diviseur pour la fréquence que nous voulons jouer. De cette façon, nous obtenons un rapport entre la fréquence souhaitée et de la fréquence de base que nous pouvons utiliser pour définir la vitesse d'un objet *groove* ~. Si nous voulons que le son sorte une octave plus haut, nous voulons que le ratio soit de 2,0; une octave plus bas, il devrait être de 0,5. La deuxième valeur sortant du fichier *coll* est formatée avec un préfixe défini par l'objet *prepend* et envoyée à *groove* ~ pour sélectionner le buffer ~ approprié à jouer. Les valeurs de boucle sortent sous forme de données numériques, fixant les points de début et de fin de boucle et envoyant une valeur de boucle dans une boîte *int* qui est déclenchée chaque fois qu'une note est jouée. Les valeurs de note-on (c'est-à-dire dont la vélocité est supérieure à 0) transmises par l'objet *stripnote* déclenchent le message de **boucle**, fixent la valeur de l'objet *sig* ~ et relancent l'échantillon en envoyant un 0 dans l'objet *groove* ~. Les objets note-off fixent l'état de la **boucle** à 0, permettant à *groove* ~ de lire l'échantillon jusqu'à la fin, puis de s'arrêter.

Échantillonneurs multi-timbraux et keymaps

- Examinez la logique du patcheur étiquetée 2 à droite du patcheur du didacticiel. Comme dans le dernier tutoriel, un objet *poly* est utilisé pour router les données MIDI vers un certain nombre d'instances d'une seule abstraction, cette fois appelée **samplervoice** ~. Cependant, les valeurs issues de l'objet *poly* font plus que définir l'allocation des voix.

- Baissez le curseur de *gain* ~ dans la zone de patcheur 1 et augmentez celui dans la zone de patcheur 2. À l'aide d'un clavier MIDI connecté, jouez quelques notes sur toute l'étendue du clavier. Remarquez que, selon les notes que vous jouez, différents échantillons sont entendus. Jouez quelques accords. Remarquez que vous pouvez avoir jusqu'à quatre notes jouées simultanément. Regardez la boîte de *message* attachée à l'objet *funbuff*:

```
0 1,  
41 2,  
48 3,  
53 4,  
68 5,  
96 6
```

L'objet *funbuff* est chargé avec ces valeurs pour les utiliser comme **mapping de touches** pour un échantillonneur multi-timbral. Toute note MIDI dont la hauteur est comprise entre 0 et 40 déclenchera l'échantillon 1 (tel que défini dans le fichier *coll*); les hauteurs comprises entre 41 et 47 déclenchent l'échantillon 2; etc.

- Double-cliquez sur l'une des abstractions nommées **samplervoice** ~ et examinez la logique de patcheur à l'intérieur. Remarquez qu'elle ressemble à la logique du patcheur dans la zone 1 du

patcheur principal, avec l'ajout de quelques objets pour gérer la vélocité MIDI et mettre à l'échelle la sortie d'amplitude de l'objet *groove* ~.

La vélocité MIDI des notes entrantes est divisée par **127**. pour la mettre à l'échelle entre **0** et **1**. Elle est ensuite multipliée par elle-même, créant une mise à l'échelle exponentielle dans laquelle des valeurs les plus élevées sur le continuum de la vélocité MIDI produisent des augmentations de volume bien plus importantes que les valeurs plus faibles. Ceci simule le comportement de circuits de volume logarithmique (tels que des faders de mixeur) dans les équipements audio analogiques.

- Revenez au patcheur principal et cliquez sur la boîte *toggle* intitulée "Lire une séquence". Double-cliquez sur l'objet du *patcheur* nommé **sequence** pendant que vous vous éclatez sur la séquence qui sort de notre échantillonneur.

Le sub-patch **sequence** contient un objet *seq*, un *midiflush* et un *midiparse*. Ces objets chargent un fichier MIDI et sortent les octets bruts en réponse à des messages **start** et **stop** (*seq*), arrêtent toutes les notes qui sonnent dans un flux d'octets MIDI en réponse à un **bang** (*midiflush*), puis analysent et extraient les paires hauteur/vélocité d'un flux MIDI afin qu'elles puissent être utilisés ailleurs (*midiparse*). Cela permet à notre impressionnante séquence MIDI d'être jouée par la logique de l'échantillonneur dans le patch principal.

Sommaire

Des échantillonneurs contrôlables par MIDI peuvent être créés en utilisant les objets *buffer* ~ et *groove* ~ de MSP. Les différents paramètres des données de l'échantillonneur (points de boucle, nom de l'échantillon, touche de base) peuvent être stockés dans des fichiers *coll* pour un accès facile, de sorte que vous pouvez facilement changer d'échantillon en fonction des événements MIDI dans la même logique de patcheur MSP.