

17-Lecture avec des boucles

Comme nous l'avons vu dans les deux derniers tutoriels d'échantillonnage, il existe une variété d'objets permettant d'accéder aux données d'un objet MSP *buffer ~*. Ce tutoriel en examine un autre spécialisé dans le travail avec des sons en boucle: *groove ~*.

Jouer des échantillons avec *groove ~*

L'objet *groove ~* est l'un des objets les plus polyvalents pour lire des sons à partir d'un *buffer ~*. Vous pouvez spécifier le *buffer ~* qu'on va lire, le point de départ, la vitesse de lecture (en avant ou en arrière), ainsi que les points de début et de fin pour une boucle répétitive dans l'échantillon. Comme pour les autres objets qui lisent à partir d'un *buffer ~*, *groove ~* accède au *buffer ~* à distance, sans cordons de connexion, en partageant son nom.

Jetez un oeil à l'objet *groove ~* dans le patcheur du tutoriel. L'objet *groove ~* possède trois entrées qui acceptent à la fois des signaux et des messages Max standard. L'entrée de gauche interprète un signal comme sa **vitesse** de lecture et un message Max de nombre à virgule flottante comme sa position de lecture. Cela permet un contrôle au niveau de l'événement du point de lecture (pour sauter à une autre zone de l'échantillon) ainsi qu'un signal variable dans le temps contrôlant la vitesse de lecture du son. Les deuxième et troisième entrées vous permettent de spécifier des points de début et de fin à utiliser en tant que boucles dans le *buffer ~*.

L'objet *sig ~* est un objet MSP qui émet un signal constant basé sur une entrée numérique à virgule flottante. Ce signal détermine la vitesse de lecture de *groove ~*. Un signal de **1** provoque une lecture en avant à vitesse normale; un signal de **0,5** divise la vitesse par deux (et diminue la hauteur de l'échantillon d'une octave); un signal de **-2** joue l'échantillon à double vitesse en sens inverse, et ainsi de suite. Bien que notre tutoriel utilise des signaux à valeur constante pour la vitesse de lecture (fournis par l'objet *sig ~*), il n'y a aucune raison pour que nous ne puissions pas utiliser *line~*, *cycle ~* ou un autre objet MSP comme signal de contrôle de la vitesse de lecture. .

L'objet *groove ~* comprend également un certain nombre de messages dans son entrée gauche pour contrôler son comportement. Le message **loop 1** active le bouclage, ce qui signifie que lorsque l'objet *groove ~* atteint la fin de l'échantillon (ou le temps spécifié dans la troisième entrée), il revient instantanément au début de l'échantillon (ou au point spécifié dans la deuxième entrée). De plus, vous pouvez changer le *buffer ~*, auquel l'objet *groove ~* accède à tout moment, comme mémoire d'échantillon, en envoyant le message **set** suivi du nom d'un objet *buffer ~* valide – en fait, vous pouvez faire cela avec n'importe quel objet MSP accédant à *buffer ~* – y compris *index ~* et *play~*. Dans le patcheur du tutoriel, un objet *umenu* est chargé avec les noms des trois objets *buffer ~* que nous avons dans notre patcheur.

Jouer quelques boucles

- Activez l'audio avec l'objet *toggle* et réglez le volume de sortie du patch à l'aide de la boîte de *nombre* à virgule flottante à droite du didacticiel (intitulé **Amplitude**). Cliquez sur les différents cercles de l'objet *preset* pour lire les échantillons de différentes manières.

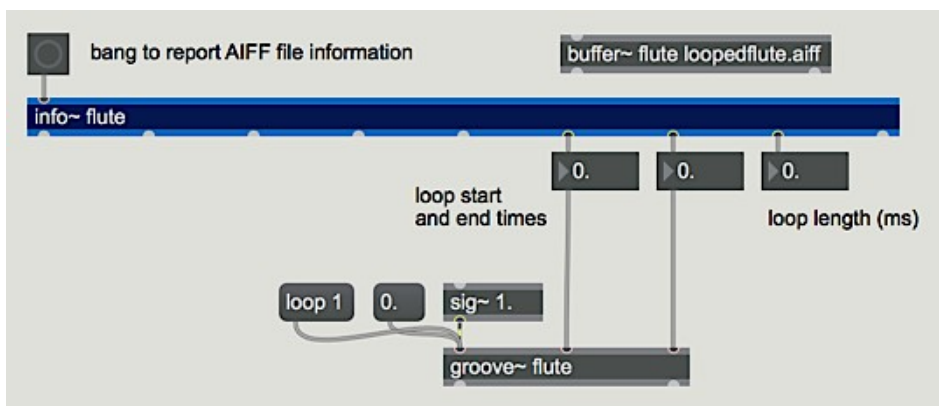
Le premier préréglage fonctionne simplement comme un bouton «Off». Les trois présélections suivantes jouent les trois objets *buffer ~* à la vitesse normale, sans boucle. Les autres préréglages présentent une variété de possibilités sonores en utilisant différentes vitesses de lecture sur différents extraits des fichiers en mémoire tampon, avec ou sans bouclage.

- Vous voudrez peut-être expérimenter avec vos propres réglages en modifiant directement les objets de l'interface utilisateur.

Détail technique: dans la conception traditionnelle des échantillonneurs, le système qui lit les échantillons a un fonctionnement très similaire à celui d'un objet `groove ~`. Typiquement (pour les échantillons instrumentaux destinés à être mis en boucle), l'échantillonneur contient une poignée d'échantillons enregistrés d'une durée de quelques secondes, chacun d'entre eux étant lié à une tonalité fondamentale spécifique, une valeur de note à laquelle il joue normalement. Les valeurs de note sans échantillons jouent la note la plus proche à laquelle un échantillon est assigné, soit légèrement plus vite, soit légèrement plus lentement en ajustant sa vitesse de lecture. Pour les notes destinées à être soutenues (par exemple, des violons ou des pads de synthétiseur), l'échantillon contient des métadonnées qui définissent des points de boucle dans l'échantillon, dans lesquels le synthétiseur boucle pendant les notes soutenues. Lorsque vous relâchez la note, l'échantillon est joué jusqu'à la fin.

Si vous souhaitez créer des boucles lisses indétectables avec `groove ~`, vous pouvez utiliser le message **loopinterp** pour activer des fondus enchaînés entre la fin d'une boucle et le début du passage suivant dans la boucle afin d'adoucir la transition vers le point de départ (voir `groove ~` page de référence pour plus d'informations sur ce message). Si le `buffer ~` contient un fichier AIFF qui a ses propres points de boucle - points établis dans un programme d'édition audio séparé - il existe un moyen d'utiliser ces points de boucle pour définir les points de boucle de `groove ~`.

L'objet `info ~` peut rapporter les points de boucle d'un fichier AIFF contenu dans un `buffer ~`, et vous pouvez envoyer ces temps de début et de fin de boucle directement dans le `groove ~`. Vous pouvez également utiliser l'objet `info ~` pour connaître la longueur (en millisecondes) de tout fichier que vous chargez dans une mémoire tampon.



Utilisation de `info ~` pour obtenir des informations sur les points de boucle d'un fichier AIFF

Résumé

L'objet `groove ~` est le moyen le plus polyvalent de lire un son à partir d'un `buffer ~`. Vous pouvez spécifier le `buffer ~` à lire, le point de départ, la vitesse de lecture (en avant ou en arrière), ainsi que les points de départ et d'arrivée d'une boucle répétitive au sein de l'échantillon. Si le `buffer ~` contient un fichier AIFF qui possède ses propres points de boucle préétablis, vous pouvez utiliser l'objet `info ~` pour obtenir ces temps de boucle et les envoyer à `groove ~`. La vitesse de lecture de `groove ~` est déterminée par la valeur du signal arrivant dans son entrée gauche. Vous pouvez définir la position actuelle du tampon de `groove ~` en envoyant une valeur de temps **float** dans l'entrée gauche.