18-table d'onde à longueur variable

Dans ce didacticiel, nous nous intéressons à l'objet *wave* ~, qui vous permet d'accéder aux données d u *buffer* ~ en tant que table d'onde pour un oscillateur. Cela vous permet de charger des échantillons et de les utiliser comme formes d'onde auxquelles on accède en fonction de la fréquence de lecture. Vous pouvez utiliser ces tables d'onde pour construire des oscillateurs basés sur des échantillons pour des patchs de synthèse plus complexes.

Utilisez n'importe quelle partie d'un *buffer* ~ comme une table d'onde: *wave* ~

Comme nous l'avons montré dans un précédent tutoriel, l'objet *cycle* ~ peut échantillonner un *buffer* ~ comme une table d'onde qu'il lit de manière répétée pour jouer un son qui se répète périodiquement. L' objet *wave* ~ est une extension de cette idée; il vous permet d'utiliser n'importe quelle section d'un *buffer* ~ comme une table d'onde avec un contrôle détaillé des points d'arrivée.

Les points de départ et d'arrivée de *buffer* ~ sont déterminés par le nombre ou le signal reçu dans les entrées centrale et droite de l'objet *wave* ~. Lorsqu'un signal reçu dans l'entrée gauche de l'objet *wave* ~ passe de **0** à **1**, *wave* ~ envoie le contenu du *buffer* ~ du point de départ spécifié au point d'arrivée. L'objet *phasor* ~, qui passe de 0 à 1 de manière répétée, est le choix évident comme signal d'entrée pour l'entrée gauche de *wave* ~.

Dans une implémentation standard de synthèse de table d'onde, la table d'onde (16K échantillons dans le cas de *cycle* ~, ou une section de n'importe quelle longueur dans le cas de *wave* ~) serait un cycle unique d'une forme d'onde, et la fréquence du l'objet *cycle* ~ (ou le *phasor*~ pilotant le *wave*~) déterminerait la fréquence fondamentale du son. Dans le cas de *wave* ~, cependant, la table d'onde pourrait contenir pratiquement n'importe quoi (une phrase parlée entière, par exemple).

wave~ donne des résultats plutôt imprévisibles par rapport à certaines des idées de génération de sons plus traditionnelles présentées jusqu'à présent, mais avec un peu d'expérimentation, vous pouvez trouver une grande variété de timbres en utilisant *wave* ~. Dans ce patch de tutoriel, vous verrez quelques façons de lire le contenu d'un *buffer* ~ avec *wave* ~.

Synthèse avec un segment de son échantillonné

Le patch de tutoriel est conçu pour vous permettre d'essayer trois façons différentes de piloter *wave* ~ (étiquetées **1**, **2** et **3**) avec deux échantillons audio différents (étiquetés **A** et **B**): avec un signal de rampe répétitif (*phasor* ~), une sinusoïde (*cycle* ~), ou une rampe unique (*line* ~). La partie inférieure du patcheur est consacrée à l'implémentation de base de *wave* ~, et la partie supérieure de la fenêtre contient les trois méthodes de lecture de la table d'ondes. Tout d'abord, regardons la moitié inférieure de la fenêtre.

• Activer l'audio. Réglez l'amplitude à un niveau confortable. Cliquez sur la boîte de **message** étiquetée **A**. Ceci demandera à l'objet *wave* ~ d'utiliser les mots nommés du *buffer* ~ et fixera la longueur de la table d'onde à la longueur de l'échantillon. Nous sommes maintenant prêts à lire la table d'onde.

• Choisissez "Lire en avant" dans le menu contextuel au milieu de la fenêtre. Cela ouvrira la première entrée de signal du sélector~, permettant ainsi à *wave* ~ d'être contrôlé par l'objet *phasor* ~ du circuit situé en haut à gauche (libellé **1**).

• Réglez la boîte de *nombre* marquée «Range» sur **1**. Ce qui définit l'amplitude du *phasor* ~, et détermine la fraction de la table d'ondes qui sera utilisée. Réglez la boîte de *nombre* marquée

«Frequency» sur **2**. Le *phasor ~* passe maintenant de 0 à 1 deux fois par seconde. Vous devriez donc entendre la lecture de *wave ~* dans le *buffer ~* toutes les demi-secondes.

• Essayez différentes valeurs de fréquence sub-audio pour *phasor~*, afin de lire *buffer~* à différentes vitesses. Vous pouvez modifier la partie de *buffer ~* qui est lue, soit en modifiant la valeur 'Range', soit en modifiant les temps de début et de fin de *wave ~* au bas du patcheur. Essayez également les fréquences audio du *phasor ~*.

Notez que le taux du *phasor ~* n'a souvent aucune relation évidente avec la hauteur perçue, parce que le contenu de la table d'onde ne représente pas un cycle unique d'une forme d'onde. De plus, une répétition aussi rapide d'un segment arbitrairement sélectionné d'un échantillon complexe a une très fortes probabilité de produire des fréquences largement supérieures au taux de Nyquist, qui seront repliées dans la plage audible de manière imprévisible.

• Cliquez sur la boîte de *message* étiquetée **B** pour référencer *wave ~* et à l'objet *buffer ~* nommé **chords**.

Cela change le contenu de la table d'onde (parce que *wave ~* accède maintenant à un *buffer ~* différent) et fixe la valeur de la boîte de *nombre* «End time» à la longueur du fichier ***sacre.aiff***. Remarquez un petit truc de programmation supplémentaire - illustré dans l'exemple ci-dessous - utilisée pour empêcher l'utilisateur de saisir des temps de début et de fin inappropriés pour *wave ~*.

Chaque fois que le temps de début ou de fin est modifiée, il révise les limites de l'autre boîte de nombre

• Avec ce nouveau *buffer ~*, faites d'autres expériences en lisant des segments de différentes longueurs du *buffer ~* à différentes vitesses.

Utiliser *wave ~* comme fonction de transfert

Examinez la section de la logique de patcheur étiquetée **2**. L'objet *buffer ~* peut également être utilisé comme une table de recherche à laquelle d'autres objets peuvent accéder de différentes manières. Dans le prochain tutoriel, nous verrons cela de manière plus traditionnelle, mais avec l'objet *wave ~*, vous pouvez générer des timbres intéressants en introduisant une forme d'onde simple dans l'objet comme valeur d'index. Pour utiliser *wave ~* de cette manière, nous devons mettre à l'échelle et décaler le *cycle ~* entrant afin qu'il envoie des valeurs de signal de 0 à 1.

• Réglez les temps de début et de fin de *wave ~* de manière rapprochée, afin que seules quelques millisecondes de son soient utilisées pour la table d'onde. Choisissez "Read back and forth" dans le menu déroulant au milieu de la fenêtre. Cela ouvre la deuxième entrée de signal du sélecteur, permettant ainsi à *wave ~* d'être contrôlé par l'objet *cycle ~*.

• Dans un premier temps, réglez la boîte de *nombre* «Range» sur une valeur très faible, telle que **0,01**, afin de limiter l'amplitude de l'objet *cycle ~*. De cette façon, *cycle ~* utilisera un très petit segment de la table d'onde comme fonction de transfert. Réglez la fréquence de *cycle ~* sur **220** Hz. Vous entendrez probablement un son riche avec une fréquence fondamentale de 220 Hz. Faites

glisser le curseur sur la boîte de *nombre* «Range» pour modifier l'amplitude de l'onde cosinusoïdale; le timbre changera en conséquence. Vous pouvez également expérimenter différentes longueurs de table d'onde en modifiant les temps de début et de fin de *wave* ~. Les fréquences sub-audio de l'objet *cycle* ~ produiront des effets inhabituels analogues à un vibrato lorsqu'il balayera la table d'onde d'avant en arrière.

Jouez le segment comme une note

Regardez la troisième section de la logique de patcheur en haut, étiquetée **3**. Parce que *wave* ~ accepte n'importe quel signal d'éntrée dans la plage 0 à 1, vous pouvez lire la table d'ondes une seule fois en envoyant à *wave* ~ un signal de rampe de 0 à 1 (ou inversement, de 1 à 0). D'autres objets, tels que *play* ~ et *groove* ~, sont mieux adaptés à cette fin, mais cela est néanmoins possible avec *wave* ~.

• Choisissez «Read once» dans le menu déroulant situé au milieu de la fenêtre. Ceci ouvre la troisième entrée de signal du *selector* ~, permettant ainsi à *wave* ~ d'être contrôlé par l'objet *line* ~. Définissez les temps de début et de fin de votre table d'onde, réglez la boîte de *nombre* «Duration» sur **1 000** et cliquez sur *button* pour parcourir la table d'onde en une seconde. Expérimentez avec les deux objets *buffer* ~, en utilisant différentes longueurs et durées de tables d'onde.

Changer la table d'onde dynamiquement

L'objet *cycle* ~ dans la partie droite de la fenêtre du Patcher est utilisé pour ajouter un changement de position sinusoïdal à la table d'onde. Comme l'onde cosinusoïdale monte et descend, les temps de début et de fin de la table d'onde augmentent et diminuent. De ce fait, la table d'ondes change constamment de position dans le *buffer* ~, de manière sinusoïdale. Soniquement, cela produit une sorte de vibrato unique, non pas de fréquence fondamentale mais de timbre. La longueur de la table d'onde et la vitesse à laquelle elle est lue restent inchangées, mais le contenu de la table d'onde change constamment.

• Réglez le «Shift Amount» sur **0,35**, et réglez le «Shift Rate» sur **6**. Réglez le temps de début de la table d'ondes à **102** et le temps de fin à **109**. Cliquez sur la boîte de *message* pour référencer *wave* ~ à l'objet *buffer*~ **chords** . Choisissez "Read forward" dans le menu contextuel. Réglez la fréquence du *phasor* ~ à un taux audio tel que **110** et réglez sa plage à **1**. Vous devriez entendre un changement de timbre semblable à un vibrato à un taux de 6 Hz. Expérimentez en faisant varier le taux de décalage et de la quantité de décalage. Lorsque vous avez terminé, cliquez sur le *toggle* pour désactiver l'audio.

Résumé

Tout segment du contenu d'un *buffer* ~ peut être utilisé comme table d'onde pour l'objet *wave* ~. Vous pouvez lire la table d'onde en envoyant un signal à *wave* ~ qui va de 0 à 1. Ainsi, en connectant la sortie d'un objet *phasor* ~ à l'entrée de *wave* ~, vous pouvez lire la table d'onde de manière répétée à un taux audio ou sub-audio. Vous pouvez également mettre à l'échelle et décaler la sortie d'un objet *cycle* ~ de manière à ce qu'elle se situe dans la plage 0 à 1, et l'utiliser comme entrée de *wave*~. Cela traite la table d'onde comme une fonction de transfert et résulte en une synthèse de mise en forme d'onde. La position de la table d'onde dans le *buffer* ~ peut être variée dynamiquement - en ajoutant un décalage sinusoïdal aux temps de début et de fin de *wave* ~, ce qui donne lieu à des types uniques de modulation de timbre.