

## 16-Lecture simultanée

Une partie de la logique derrière la façon dont MSP travaille avec les échantillons est que plusieurs objets dans un patcheur Max peuvent accéder à la même mémoire d'échantillon d'un seul objet *buffer ~*. Cela facilite l'implémentation de la **polyphonie** dans un patcheur de lecture d'échantillon, comme le montre ce didacticiel.

### Chargement d'un échantillon

Jetez un coup d'œil au patcheur du tutoriel. Comme pour le tutoriel précédent, celui-ci utilise un objet *buffer ~* (nommé **gerald**, dans ce cas), pour stocker un échantillon auquel d'autres objets peuvent accéder. Plutôt que d'enregistrer dans le *buffer ~* nous lui avons demandé de charger un échantillon (appelé "drumLoop.aif") lors du chargement du patcheur. Le message **replace** dans le *buffer ~* diffère du message **read** en ce sens qu'il redimensionne la mémoire de l'objet *buffer ~* pour accueillir l'intégralité du fichier son lu; bien que dans ce tutoriel, cela ne fasse aucune différence, les deux messages se comportent différemment une fois que vous commencez à charger et décharger des échantillons dans le même objet *buffer ~*.

- Double-cliquez sur l'objet *buffer ~* pour afficher le contenu de l'échantillon appelé **gerald**.



*un échantillon appelé gerald*

Si nous jetons un coup d'œil à l'échantillon chargé, nous constatons qu'il semble contenir quatre sons espacés dans le temps d'environ une seconde. Dans ce patcheur, cet échantillon est accessible par une série de trois objets *play ~* différents, tous nommés d'après notre objet *buffer ~* ... ces trois objets *play ~* sont intégrés à la logique du patcheur pour lire différentes parties de l'échantillon.

### Une simple boîte à rythmes

- Activez l'audio pour le patch de tutoriel et montez le curseur de *gain ~*. Cliquez sur les objets *button* du patch et écoutez le résultat. Ces objets *button* déclenchent des messages qui ordonnent aux différents objets *play ~* d'accéder aux différentes parties du *buffer ~* que nous avons chargé. Les quatre différentes rampes générées par les objets *line ~* jouent chacune une partie différente d'une seconde du *buffer ~*. En raison de la façon dont nous avons conçu le fichier audio que le *buffer ~* a chargé, ces sons correspondent aux quatre sons que nous avons vus dans l'affichage de la forme d'onde de *buffer ~*: quatre coups de batterie d'une boîte à rythmes TR-808 ... une grosse caisse, une caisse claire, un charleston fermé et un charleston ouvert.

Comme nos trois objets *play ~* sont capables d'accéder au même contenu de *buffer ~*, ils peuvent être déclenchés simultanément et créer un son polyphonique. Le reste de la logique du patcheur simule une boîte à rythmes aléatoire très simple, qui utilise des objets aléatoires pilotés par un *metro* pour déclencher des parties du *buffer ~* en fonction des boîtes de **messages**. La boîte à rythmes utilise des objets *random* pour définir la probabilité d'un événement sonore pour la grosse caisse (une chance sur trois), la caisse claire (une chance sur quatre) et les charleston (deux chances

sur trois, également réparties entre les deux sons).

- Cliquez sur la boîte *toggle* en haut du patcheur pour lancer le *metro*. Une fois que la boîte à rythmes démarre, remarquez qu'il est possible d'entendre simultanément une grosse caisse, une caisse claire et un des échantillons de charleston, même si ces trois sons se trouvent dans le même *buffer* ~. La façon dont nous avons construit notre patcheur nous place dans l'impossibilité d'entendre un charley ouvert et un charley fermé en même temps; ces deux *triggers* d'échantillons vont au même objet *play* ~, qui ne peut faire sonner qu'une seule «voix» de notre boîte à rythmes. En conséquence, ils s'interrompent l'un l'autre s'ils se déclenchent trop rapidement.

## Résumé

L'architecture d'échantillonnage de MSP permet à un nombre quelconque d'objets de lecture d'échantillons d'accéder au contenu du même objet *buffer* ~. Par conséquent, vous pouvez utiliser plusieurs objets *play* ~ pour accéder simultanément à des données d'échantillons afin de créer une texture polyphonique. La création de fichiers audio contenant des échantillons à intervalles de temps réguliers facilite la mise en place de «banques» d'échantillons au sein d'un même fichier, accessible depuis MSP en lisant à différents points d'échantillon dans un *buffer* ~.