

3-patches Max et réseau de signaux MSP

introduction

Les objets Max communiquent en échangeant des messages par le biais de cordons de connexion. Ces messages sont envoyés à un moment précis, soit en réponse à une action de l'utilisateur (un clic de souris, une note MIDI jouée, etc.) ou parce que l'événement a été programmé pour se produire (par *metro*, *delay*, etc.).

Les objets MSP sont connectés par des cordons de connexion de manière similaire, mais leur intercommunication est différente sur le plan conceptuel. Plutôt que d'établir un chemin pour l'envoi des messages, les connexions MSP établissent une relation entre les objets connectés et cette relation est utilisée pour calculer les informations audio nécessaires à un instant donné. Cette configuration d'objets MSP est appelée réseau de signaux.

L'exemple suivant illustre la distinction entre un patch Max dans lequel des messages sont envoyés et un réseau de signaux dans lequel une relation en cours est établie.



Les messages Max se produisent à un moment donné. Les objets MSP sont en communication constante.

Dans l'exemple Max à gauche, la boîte de *nombres* ne connaît pas le nombre 0.75 stocké dans l'objet *float*. Lorsque l'utilisateur clique sur le bouton, l'objet *float* envoie sa valeur stockée. Ce n'est qu'alors que la boîte de *nombres* reçoit, affiche et envoie le nombre **0,75**. Dans l'exemple MSP à droite, cependant, chaque prise connectée au réseau de signaux contribue constamment à l'équation avec sa valeur actuelle. Ainsi, sans qu'aucun message Max spécifique ne soit envoyé, l'objet *~ reçoit la sortie des deux objets *sig~*, et tout objet connecté à la sortie de *~ recevra le produit **0,75**.

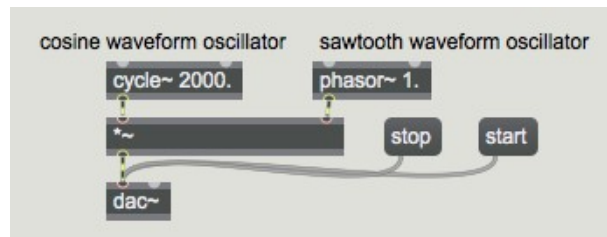
Une autre façon de penser à un réseau de signaux MSP consiste à utiliser une partie d'un patch qui (audio) fonctionne à un débit plus rapide que Max. Max, et vous, l'utilisateur, ne pouvez affecter directement cette portion de signal du patch que toutes les millisecondes. Ce qui se passe entre ces intervalles de millisecondes est calculé et effectué par MSP. Si vous pensez à un réseau de signaux de cette manière - comme un patch très rapide -, il est toujours logique de considérer les objets MSP comme "envoyant" et "recevant" des messages (même si ces messages sont envoyés plus vite que Max ne peut les voir). Nous continuerons donc à utiliser la terminologie standard de Max, telle que *send*, *receive*, *input* et *output* pour les objets MSP.

Taux audio et taux de contrôle

L'unité de temps de base pour programmer des événements dans Max est la milliseconde (0,001 seconde). Ce débit (1 000 fois par seconde) est généralement assez rapide pour tout type de contrôle que l'on peut vouloir exercer sur des périphériques externes tels que des synthétiseurs ou des effets visuels tels que des films QuickTime.

L'audio numérique doit cependant être traité à un rythme beaucoup plus rapide – généralement 44100 fois par seconde par canal audio. La façon dont MSP gère cela consiste à calculer, en permanence, tous les nombres qui seront nécessaires pour produire les prochaines millisecondes

d'audio. Ces calculs sont effectués par chaque objet, en fonction de la configuration du réseau de signaux.



Un oscillateur (*cycle ~*) et un amplificateur (** ~*) commandé par un autre oscillateur (*phasor ~*)

Dans cet exemple, un oscillateur à forme d'onde cosinusoidale d'une fréquence de 2000 Hz (l'objet *cycle~*) voit son amplitude mise à l'échelle (chaque échantillon est multiplié par un certain nombre dans l'objet **~*), puis envoyé au convertisseur numérique-analogique (*dac~*). Au cours de chaque seconde, la sortie de l'onde en dents de scie (sub-audio) de l'objet *phasor~* envoie une rampe continue de valeurs croissantes de 0 à 1. Ces nombres croissants seront utilisés comme opérande de droite dans le **~* pour chaque échantillon de la forme d'onde audio, et le résultat sera que le son de 2000 Hz passera de manière linéaire du silence à la pleine amplitude à chaque seconde. Pour chaque milliseconde d'audio, MSP doit produire environ 44 valeurs d'échantillon (en supposant un taux d'échantillonnage audio de 44 100 Hz). Il doit donc chercher la valeur appropriée dans chaque oscillateur et multiplier ces deux valeurs pour produire l'échantillon de sortie.

Même si de nombreux objets MSP acceptent les valeurs d'entrée exprimées en millisecondes, ils calculent les échantillons à une fréquence d'échantillonnage audio. Les messages Max voyagent beaucoup plus lentement, à ce que l'on appelle souvent un taux de **contrôle**. Il est peut-être utile de penser qu'il existe effectivement deux taux d'activité différents: le taux de **contrôle** plus lent du planificateur de Max et le taux **d'échantillonnage** audio plus rapide.

Remarque: puisque vous pouvez spécifier le temps dans Max en millisecondes à virgule flottante, la résolution du planificateur varie en fonction de la fréquence de son exécution. Le taux de contrôle exact est défini par un certain nombre de paramètres MSP que nous présenterons sous peu. Toutefois, il est beaucoup moins efficace de "traiter" l'audio en utilisant les fonctions "de contrôle" exécutées dans le planificateur que d'utiliser les objets audio spécialisés de MSP.

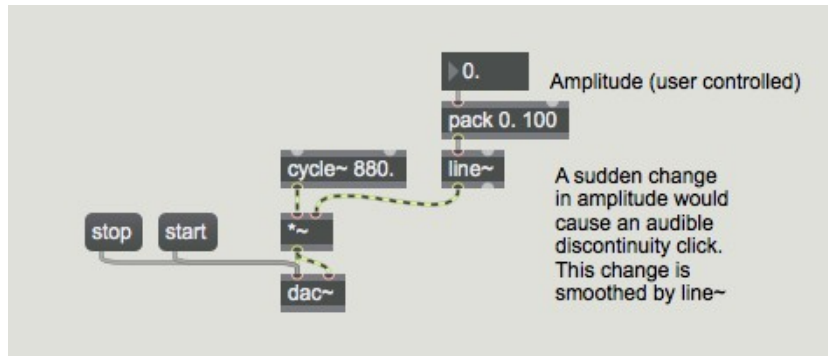
Le lien entre Max et MSP

Certains objets MSP existent spécifiquement pour fournir un lien entre Max et MSP - et pour effectuer la conversion entre le taux de contrôle et le taux audio. Ces objets (tels que *sig~* et *line ~*) prennent les messages Max dans leurs entrées, mais leurs sorties se connectent au réseau de signaux; ou inversement, certains objets (tels que *snapshot~*) se connectent au réseau de signaux et peuvent afficher (mais seulement aussi fréquemment qu'une fois par milliseconde) la ou les valeurs présentes en un point particulier du réseau de signaux.



Fournir un message Max au réseau de signaux ou obtenir un message Max d'un signal

Ces objets sont très importants car ils donnent à Max, et à vous utilisateur, le contrôle direct de ce qui se passe dans le réseau de signaux.



Contrôle de l'interface utilisateur sur l'amplitude du signal

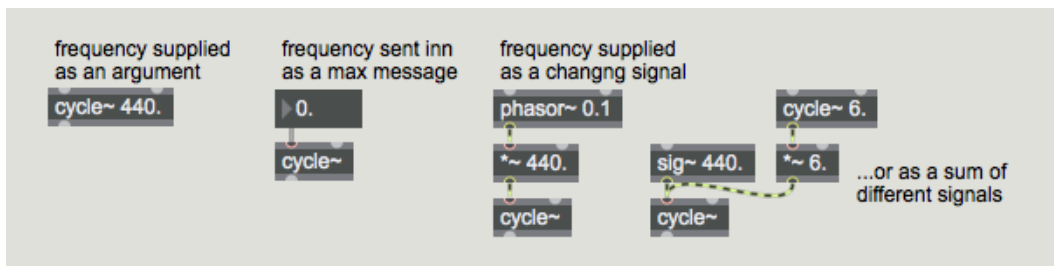
Certaines entrées d'objet MSP acceptent à la fois les signaux et les messages Max. Ils peuvent être connectés en tant que partie d'un réseau de signaux et ils peuvent également recevoir des instructions ou des modifications via des messages Max.

Par exemple, l'objet *dac~* (convertisseur numérique-analogique), pour la lecture du signal audio, peut être activé et désactivé avec les messages Max **start** et **stop**.



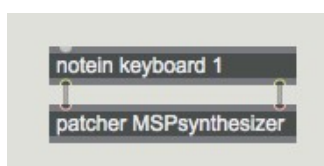
Certains objets MSP peuvent recevoir des signaux audio et des messages Max dans la même entrée.

Et l'objet *cycle~* (oscillateur) peut recevoir sa fréquence sous forme de message Max **float** ou **int**, ou bien il peut recevoir sa fréquence d'un autre objet MSP (bien qu'il ne puisse pas faire les deux en même temps, car l'entrée audio peut être considérée comme fournissant constamment des valeurs qui annuleraient immédiatement l'effet du message float ou int).



Certains objets MSP peuvent recevoir des messages Max ou de signaux dans la même but.

Vous voyez donc qu'un patch Max (ou un subpatch) peut contenir à la fois des objets Max et des objets MSP. Pour une organisation claire, il est souvent utile d'encapsuler un processus entier, tel qu'un réseau de signaux, dans un subpatch afin qu'il puisse apparaître en tant qu'objet unique dans un autre patch Max.



L'encapsulation peut clarifier les relations dans un patch Max

Limites de MSP

De la discussion précédente, il est évident que le traitement audio numérique nécessite beaucoup de «calculs». L'ordinateur doit produire des dizaines de milliers d'échantillons par seconde et par canal de son. Et chaque échantillon peut nécessiter de nombreux calculs arithmétiques, en fonction de la complexité du réseau de signaux. Et pour produire de l'audio *en temps réel*, les échantillons doivent être calculés au moins aussi vite qu'ils sont joués.

La synthèse sonore en temps réel d'une telle complexité sur un ordinateur personnel à usage polyvalent était pratiquement exclue jusqu'à l'introduction de processeurs suffisamment rapides, tels que le PowerPC. Même si la vitesse et la capacité des ordinateurs contemporains ont largement dépassé le PowerPC, ce type de traitement des nombres requiert une grande partie de l'attention du processeur. Il est donc important de savoir qu'il existe des limites à ce que votre ordinateur peut faire avec MSP. (Aucun processeur audio dédié ne peut nous secourir comme la carte graphique qui accélère le traitement vidéo.)

Contrairement à un synthétiseur MIDI, dans MSP vous avez la possibilité de concevoir quelque chose qui est trop compliqué pour que votre ordinateur puisse le calculer en temps réel. Le résultat peut être une distorsion audio, un ordinateur très peu réactif ou, dans des cas extrêmes, un crash.

En raison de la variation des performances du processeur entre les ordinateurs et de la grande variété des configurations possibles du réseau de signaux, il est difficile de dire avec précision quelle complexité du traitement audio MSP peut ou ne peut pas gérer. Voici quelques principes généraux:

- Plus le CPU de votre ordinateur est rapide, meilleures seront les performances de MSP.
- Un disque dur rapide et une connexion rapide amélioreront les entrées / sorties des fichiers audio.
- La désactivation des processus en arrière-plan (comme le chat Internet) améliorera les performances.
- Réduire le taux d'échantillonnage audio réduira le nombre de nombres que MSP doit calculer pour une quantité donnée de son, améliorant ainsi ses performances (bien qu'un taux d'échantillonnage plus faible entraîne une dégradation de la réponse en haute fréquence). Le contrôle du taux d'échantillonnage audio est traité dans le chapitre *Input* et *Output* audio.

Lorsque vous concevez vos instruments MSP, vous devez garder à l'esprit que certains objets nécessitent des calculs plus intensifs que d'autres. Un objet qui n'effectue que quelques opérations arithmétiques simples (telles que *sig~*, *line~*, *+~*, *floating-point number box*, **~* ou *phasor~*) est peu coûteux en calcul. (Cependant, */~* est beaucoup plus coûteux.) Un objet qui recherche un nombre dans une table de fonctions et interpole entre les valeurs (comme *cycle~*) ne nécessite que quelques calculs et n'est donc pas trop coûteux. Les objets les plus coûteux sont ceux qui doivent effectuer de nombreux calculs par échantillon: les filtres (*reson~*, *biquad~*), les analyseurs spectraux (*fft~*, *ifft~*) et les objets tels que *play~*, *groove~*, *comb~* et *tapout~* lorsqu'un de leurs paramètres est contrôlé par un signal continu. Les questions d'efficacité sont traitées plus en détail dans le *didacticiel MSP*.

*Remarque: pour connaître la durée de traitement de votre patch par le processeur, consultez **CPU Utilization** du processeur dans le moniteur de processeur DSP. Vous le*

trouvez dans le menu Extras.

Avantages de MSP

Votre ordinateur portable ou votre ordinateur de bureau est un ordinateur à usage général, et non un ordinateur de traitement du son spécialement conçu, tel qu'un échantillonneur ou un synthétiseur commercial. Par conséquent, vous ne pouvez généralement pas vous attendre à ce qu'il soit aussi performant. Cependant, pour les conceptions d'instruments relativement simples qui répondent à des besoins spécifiques en matière de synthèse ou de traitement, ou pour expérimenter de nouvelles idées de traitement audio, c'est un environnement pour la construction d'instruments très pratique. Voici quelques-unes des choses que MSP peut vous permettre de faire:

1. *Concevez un instrument adapté à vos besoins.* Même si vous avez beaucoup d'équipement audio, il ne peut probablement pas faire tout ce que vous pouvez imaginer. Lorsque vous avez besoin d'accomplir une tâche spécifique non disponible dans votre studio, vous pouvez la concevoir vous-même.
2. *Construisez un instrument et écoutez les résultats en temps réel.* Avec les programmes de synthèse sonore non temps réel, vous définissez un instrument qui, à votre avis, sonne comme vous le souhaitez, puis vous le compilez et testez les résultats, vous faites quelques ajustements, vous le recompiliez, etc. Avec MSP, vous pouvez entendre chaque modification que vous apportez à l'instrument au fur et à mesure que vous le construisez, ce qui rend le processus plus interactif.
3. *Établissez la relation entre le contrôle gestuel et le résultat audio.* Avec de nombreux instruments commerciaux, vous ne pouvez pas modifier les paramètres en temps réel, ou vous ne pouvez le faire qu'en programmant un ensemble complexe de contrôles MIDI. Avec Max, vous pouvez facilement connecter des données MIDI au paramètre exact que vous souhaitez modifier dans votre réseau de signaux MSP, et vous savez précisément quel aspect du son vous contrôlez avec le MIDI.
4. *Intégrez le traitement audio dans vos programmes de composition ou de performance.* Si votre travail musical consiste à concevoir des programmes de composition automatisés ou des performances assistées par ordinateur dans Max, vous pouvez désormais intégrer le traitement audio à ces programmes. Vous avez besoin de faire un fondu-enchaîné mains libres entre votre voix et un échantillon préenregistré à un moment précis de la performance ? Vous pouvez écrire un patch Max avec des objets MSP qui le fait pour vous, déclenché par un simple message MIDI.

Certaines de ces idées sont illustrées dans les didacticiels MSP.