

## 5-Test de Tonalité

*Ouvrez le tutoriel.*

### **Les objets MSP ressemblent beaucoup aux objets Max**

Les objets MSP servent à traiter de l'audio numérique (c'est-à-dire du son) qui sera lu par votre ordinateur. Les objets MSP ressemblent aux objets Max, ont des entrées et des sorties tout comme les objets Max et sont connectés entre eux par des cordons de connexion, tout comme les objets Max. Ils sont créés de la même manière que les objets Max - en plaçant simplement une boîte d'objet dans la fenêtre du Patcher et en tapant le nom souhaité - et ils coexistent parfaitement avec les objets Max dans la même fenêtre du Patcher.

### **... mais ils sont un peu différents.**

Un patcheur contenant des objets MSP interconnectés fonctionne un peu différemment d'un patcheur contenant des objets Max ordinaires.

Une façon de voir la différence est de penser que les objets MSP travaillent beaucoup plus rapidement que les objets Max ordinaires. Puisque les objets MSP doivent produire suffisamment de nombres pour générer un signal audio de haute fidélité (généralement 44 100 nombres par seconde), ils doivent travailler plus vite que le planificateur utilisé par les objets Max standard, qui tourne généralement autour de mille fois par seconde.

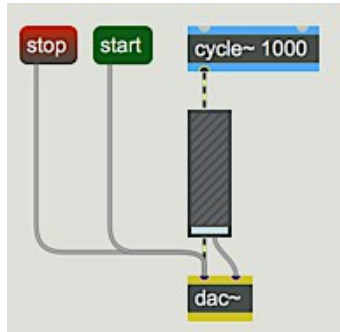
Voici une autre façon utile d'envisager la différence. Imaginez un patch d'objets MSP non pas comme un programme dans lequel les événements se produisent à des instants spécifiques (comme dans un patch Max standard), mais plutôt comme une description de la conception d'un instrument - un synthétiseur, un échantillonneur ou un processeur d'effet. C'est comme une formule mathématique, chaque objet fournissant constamment des valeurs numériques à l'objet ou aux objets connectés à sa sortie. A tout instant, cette formule a un résultat, qui est l'amplitude instantanée du signal audio. C'est pourquoi nous appelons fréquemment un ensemble d'objets MSP interconnectés un *réseau de signaux*.

Ainsi, alors qu'un patch composé d'objets Max standard reste inactif jusqu'à ce que quelque chose se produise (un clic de souris, un message MIDI entrant, le déclenchement de l'horloge d'un objet *metro*, etc.), provoquant l'envoi d'un message à un autre objet. En revanche, un réseau de signaux d'objets MSP est toujours actif (du moment où il est allumé jusqu'au moment où il est éteint), avec tous ses objets communiquant en permanence pour calculer l'amplitude appropriée du son à cet instant.

### **... alors ils ont un aspect un peu différent**

Les noms de tous les objets MSP se terminent par le caractère tilde (~). Ce caractère, qui ressemble à un cycle d'une onde sinusoïdale, sert simplement d'indicateur pour vous aider à distinguer les objets MSP des autres objets Max.

Les cordons de raccordement entre les objets MSP sont rayés. Cela vous aide à distinguer le réseau de signaux MSP du reste du patch Max.



Les objets MSP sont connectés par des cordons de connexion rayés

Jetez un coup d'œil au patcheur du tutoriel. Vous verrez que nous avons six éléments dans le cadre de notre patcheur. Trois d'entre eux devraient vous être familiers: nous avons une boîte de commentaires Max ("warning: loud") et deux boîtes de messages étiquetées **start** et **stop**. Le gros de la logique du patcheur consiste en trois objets que nous n'avons jamais vus auparavant: un objet *cycle ~*, un objet *dac ~* et quelque chose qui ressemble étrangement à un objet *slider* mais qui est en fait un objet *gain ~*. Examinons ces objets tour à tour, en commençant par le *dac ~* en bas du patcher et en revenant au sommet de la chaîne.

### Convertisseur numérique-analogique: *dac ~*

**Le convertisseur numérique-analogique** (DAC) est la partie de votre ordinateur qui traduit le flux de chiffres discrets d'un signal audio numérique en une tension continue et fluctuante qui alimentera votre haut-parleur.

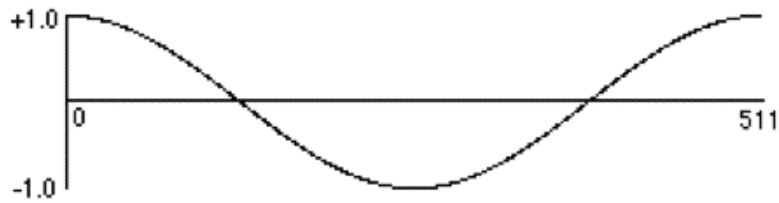
Une fois que vous avez calculé un signal numérique pour générer un son généré par ordinateur, vous devez envoyer les chiffres au DAC. Ainsi, MSP possède un objet appelé *dac ~*. Cet objet est généralement l'objet terminal de tout réseau de signaux et est nécessaire pour que les signaux générés par MSP sortent de votre ordinateur sous forme de son. L'objet *dac ~* reçoit les signaux que vous souhaitez entendre dans ses entrées. Il peut avoir autant d'entrées que de canaux de lecture audio disponibles sur votre ordinateur, bien qu'il prenne par défaut une configuration stéréo en communiquant avec les deux premiers canaux de votre matériel audio. C'est pourquoi il existe deux entrées sur le *dac ~* de notre patcheur. Si vous utilisez un matériel de sortie audio plus élaboré, vous pouvez saisir au clavier des arguments pour spécifier d'autres canaux audio.

**Important!** *dac ~* doit recevoir un signal d'amplitude non nulle pour que vous puissiez entendre quelque chose. Le *dac ~* s'attend à recevoir des valeurs de signal comprises entre -1,0 et 1,0. Les nombres qui dépassent cette plage provoqueront une distorsion lors de la lecture du son.

### Un oscillateur simple à table d'ondes: *cycle ~*

La meilleure façon de produire une forme d'onde périodique est d'utiliser *cycle ~*. Cet objet utilise la technique dite de «synthèse par table d'ondes». Il lit une liste de 512 valeurs à une vitesse donnée, en revenant au début de la liste lorsqu'il atteint la fin. Ceci simule une forme d'onde se répétant périodiquement.

Vous pouvez demander à *cycle ~* de lire une liste de valeurs que vous fournissez (sous la forme d'un fichier audio), ou si vous n'en fournissez pas, il lira sa propre table, qui représente le cycle d'une onde cosinusoidale avec une amplitude de 1. Nous vous montrerons comment fournir votre propre forme d'onde dans un prochain tutoriel ; pour l'instant, nous utiliserons la forme d'onde du cosinusoidale.



Graphique de 512 nombres décrivant un cycle d'une onde cosinusoidale d'amplitude 1

L'objet *cycle* ~ reçoit une valeur de fréquence (en Hz) dans son entrée gauche et il détermine lui-même la vitesse à laquelle il doit lire la liste afin d'émettre un signal avec la fréquence souhaitée.

**Détail technique:** pour déterminer la distance à parcourir dans la liste pour chaque échantillon consécutif, *cycle* ~ utilise la formule de base

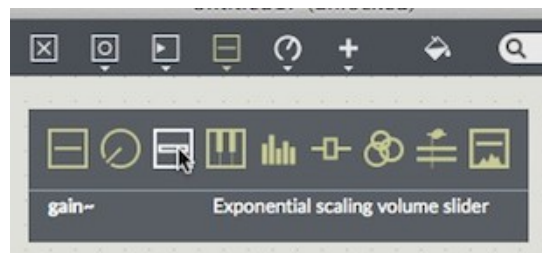
$$I = fL / R$$

où *I* est le montant à incrémenter dans la liste, *f* est la fréquence du signal, *L* est la longueur de la liste (512 dans ce cas) et *R* est le taux d'échantillonnage audio. *cycle* ~ est un 'oscillateur d'interpolation', ce qui signifie que s'il ne correspond pas exactement à un index entier dans la liste pour un échantillon donné, l'objet *cycle* ~ interpole entre les deux nombres les plus proches dans la liste pour trouver la valeur de sortie appropriée. L'interpolation dans un oscillateur à table d'onde permet d'améliorer considérablement la qualité du son. l'objet *cycle* ~ utilise une interpolation linéaire, tandis que d'autres objets MSP utilisent une interpolation polynomiale de très haute qualité (et plus onéreuse en termes de calcul).

L'objet *cycle* ~ a une fréquence par défaut de 0 Hz. Donc, pour entendre le signal, nous devons fournir une valeur de fréquence audible. Cela peut être fait avec un argument numérique comme dans le patch d'exemple, ou en envoyant un nombre dans l'entrée gauche, ou en connectant un autre objet MSP à l'entrée gauche. Si nous écoutons la sortie de ce signal avec sa fréquence réglée dans une plage audible, nous devrions entendre une onde sinusoïdale, le son le plus pur (en termes de spectre) que nous puissions produire.

### Un contrôle de volume: *gain* ~

Si nous connectons directement notre objet *cycle* ~ à un objet *dac* ~ et démarrons l'audio, nous pourrions facilement endommager nos haut-parleurs ou, plus important encore, nos oreilles. Les objets MSP qui génèrent de l'audio (tel que *cycle* ~) le font généralement dans une plage dynamique normalisée de -1 à 1, que l'objet *dac* ~ interprétera comme étant le signal le plus fort possible. Pour éviter ces dangers, il est important d'avoir un objet intermédiaire pour contrôler l'amplitude du signal qui va vers nos haut-parleurs. Le curseur *gain* ~ remplit une telle fonction:



Le curseur de *gain* ~ dans la section des curseurs de la palette d'objets

Le curseur de *gain* ~ multiplie le signal entrant par un facteur basé sur la position du curseur qui, sauf lorsqu'il atteint le niveau le plus élevé, correspond à un nombre inférieur à 1. En conséquence, l'objet *gain* ~ **atténue** presque toujours (ou réduit) le volume de ce qui entre, un peu comme un

fader sur une table de mixage analogique. L'objet *gain ~* est à plusieurs égards une version graphique d'un objet \*~, que nous examinerons dans un prochain tutoriel.

*Il est important de noter que le curseur de gain ~ «monte jusqu'à 11». En fait, il va au-delà de 12, dans la mesure où le pas le plus élevé du curseur est 157 par rapport au pas de gain unitaire de 128. Ce niveau extrême multiplie le signal par plus de 7, garantissant presque une distorsion.*

## Démarrer et arrêter le traitement du signal

Le réseau audio MSP est désactivé au démarrage de Max. Il peut être activé avec l'icône on / off dans le coin inférieur droit de toute fenêtre contenant des objets MSP. Il y a un curseur situé juste au-dessus de ce bouton qui contrôle le niveau de la fenêtre dans le mixage de sortie, ainsi qu'une paire de barres VU qui montrent le niveau du signal. Lorsque la fenêtre est ouverte, ce curseur se trouve au point de gain unitaire et peut être utilisé pour équilibrer le mixage, mais gardez à l'esprit que réduire le niveau avec ce curseur peut masquer la distorsion provoquée par des niveaux excessifs dans le patch.

Le traitement audio peut également être contrôlé par programme. L'objet *dac ~* est l'un des cinq objets MSP (avec *adc ~*, leurs cousins graphiques *ezadc ~* et *ezdac ~*, et *adstatus*) qui peuvent activer et désactiver le réseau audio MSP depuis votre patch. Il est très courant de fournir le message de démarrage un objet *loadmess* afin que l'audio commence dès que le patch est ouvert.

**Important!** L'envoi d'un message **start** ou **stop** à tout objet *dac ~* ou *adc ~* active ou désactive le traitement de tous les réseaux de signaux ouverts de tous les patcheurs Max actuellement ouverts. Donc, si vous activez l'audio dans un patcheur et que vous entendez des choses inattendues, un bon endroit où regarder est le menu **Fenêtre** de Max. Il est possible que d'autres documents ouverts qui génèrent de l'audio en utilisant MSP. Vous pouvez utiliser le message **startwindow** pour démarrer le traitement uniquement dans la fenêtre concernée.

## Ecouter la tonalité de test

La première fois que vous démarrez Max, il essaiera d'utiliser la carte son et le pilote par défaut de votre ordinateur (CoreAudio sur Macintosh ou MME sous Windows) pour l'entrée et la sortie audio. Si la sortie audio de votre ordinateur est connectée à un casque ou à un amplificateur, vous devriez entendre la sortie de MSP. Si vous n'avez aucun câble audio connecté à votre ordinateur, vous entendrez le son par le haut-parleur interne de l'ordinateur.

Pour que MSP fonctionne correctement, nous vous recommandons de démarrer les didacticiels à l'aide du matériel audio intégré de votre ordinateur. Si vous souhaitez utiliser une interface audio externe ou une carte son, reportez-vous au chapitre Entrées et sorties audio pour plus de détails sur la configuration de MSP pour fonctionner avec du matériel audio.

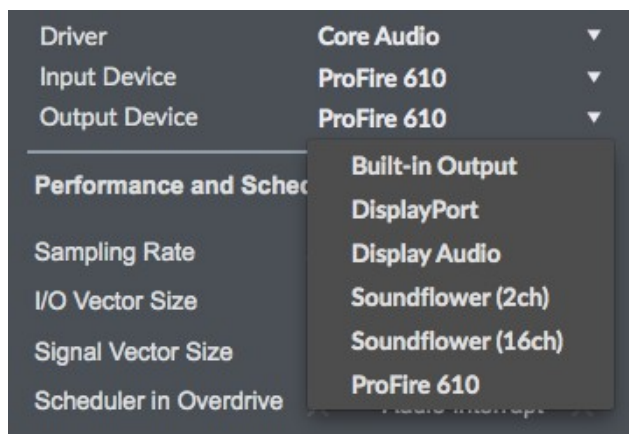
- Réglez votre amplificateur audio (ou vos haut-parleurs) sur leur réglage minimum, puis cliquez sur le message **start** dans le patch du didacticiel. Cliquez et faites glisser le curseur de *gain ~* jusqu'à ce qu'il soit placé quelque part au milieu de sa longueur. Ajustez votre amplificateur audio à un niveau confortable. Une fois que vous êtes satisfait du fonctionnement du patch, cliquez sur la boîte de message **stop** pour désactiver cette ennuyeuse tonalité de test.

## Dépannage

Si, dans cet exemple, vous n'entendez aucun son de votre ordinateur lorsque vous démarrez le *dac ~*, augmentez le curseur de *gain ~*, vérifiez le réglage de niveau de votre amplificateur ou de votre

table de mixage, ainsi que toutes vos connexions audio. Vérifiez que la sortie audio n'est pas actuellement coupée. Sur Macintosh, le niveau de sortie du son est réglé à l'aide des préférences Son de l'application Préférences Système. Sous Windows, le niveau de sortie du son est défini à l'aide de la configuration Sons et périphériques audio (Démarrer - Panneaux de configuration - Sons et périphériques audio).

Si vous n'entendez toujours rien, choisissez **Audio Status** dans le menu Options et vérifiez qu'un pilote et une sortie valides sont sélectionnés dans les menus contextuels. Si ce n'est pas le cas, choisissez-le. (Voir le tutoriel *Audio I/O* pour plus d'informations.)



*Vous pouvez toujours vérifier si les problèmes audio proviennent de votre patch ou du système audio en ouvrant le patch **Audiotester** dans le menu **Extras**. Ce patch fournit une source audio simple ainsi que des affichages de toute entrée active. Si le fait de sélectionner **Oscillator** ou **Pink noise** dans le menu source du signal ne produit pas de son, le problème se situe quelque part dans la configuration audio.*

## Résumé

Un réseau de signaux composé d'objets MSP connectés décrit un instrument audio. Le convertisseur numérique-analogique de l'instrument est représenté par l'objet *dac ~*; *dac ~* doit recevoir un signal d'amplitude non nulle (compris entre -1,0 et 1,0) pour que vous puissiez entendre quelque chose. L'objet *cycle ~* est un oscillateur à table d'ondes qui lit cycliquement une liste de 512 valeurs d'amplitude, à une vitesse déterminée par la valeur de fréquence fournie. Le traitement du signal est activé et désactivé à l'aide d'un bouton dans la fenêtre du patcheur ou en envoyant un message **start** ou **stop** à tout objet *dac ~* ou *adc ~*.

- Comme l'envoi d'un message **start** à un *dac ~* active l'audio dans tous les patchs Max ouverts qui contiennent des objets MSP, il est préférable, lorsque vous parcourez ces didacticiels, de ne les ouvrir qu'un seul à la fois. Fermez la fenêtre Patcher avant de passer au chapitre suivant.