MaxMSP – introduction/apprentissage :

les messages, le contrôle, l'audio

Tom Mays

intro

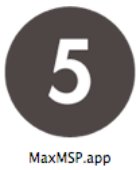
Qu'est-ce que c'est?

MaxMSP (or Max/MSP/Jitter) est un environnement graphique de programmation, audio, MIDI, et depuis quelques années, l'image – en temps-réel. A la fin des années 80 à l'Ircam, Miller Puckette a conçu et a créé MAX (en hommage à Max Mathews, pionnier de l'informatique musicale) comme interface de contrôle pour l'ancienne station audio temps réel de l'Ircam, le 4X. Plus tard, MAX a été développé tout seul comme environnement MIDI et repris par la société Opcode, avec comme développeur David Zicarelli. En 1997, Zicarelli a rajouté à MAX une librairie d'objets pour l'audio qu'il appelait MSP (d'après Miller S. Puckette, ou bien Max Signal Processing), inspiré par la station d'informatique musicale de l'Ircam et par le travail de Puckette dans son nouvel environnement temps réel, Pd. Aujourd'hui, Zicarelli continue à développer Max/MSP avec sa propre société, Cycling74 (http://www.cycling74.com).

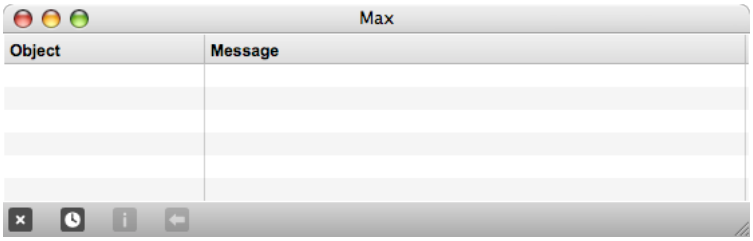
Avec Max, on construit des « patches » (programmes, configurations, fonctions, etc) en reliant des « objets » entre eux qui ont chacun une ou plusieurs fonctions précises. C'est tout…

MaxMSP et la page blanche

ouvrir l'appli

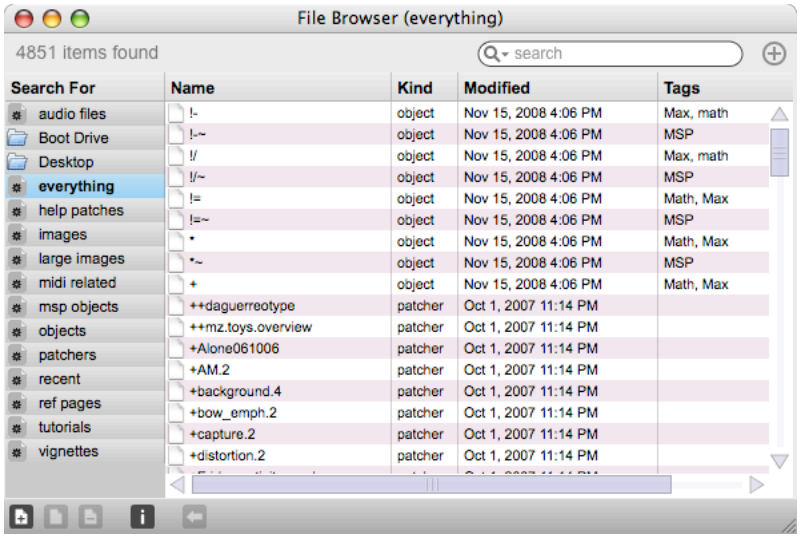
Lancer l'appli MaxMSP : 

La première fenêtre qui apparaît c'est la Max Window. C'est une fenêtre vous aller avoir l'habitude de toujours garder visible, au moins une partie, parce que c'est ici où les messages d'erreur et d'autres informations vont s'afficher.



File Browser

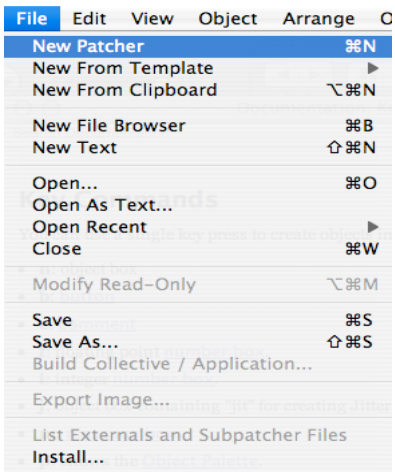
Faire *New File Browser* dans le menu *File* (ou « commande B »)



Nouveau dans la version 5, le *File Browser* permet de rechercher des éléments par critère, et de glisser/déposer des éléments qui peuvent être glissé/déposé…

le Patcher

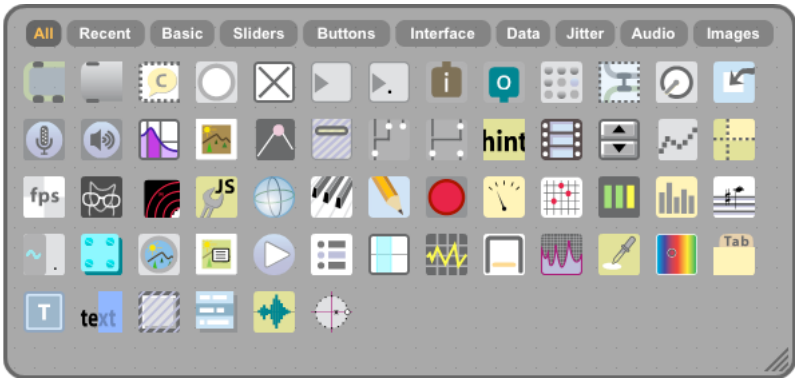
Aller dans le menu *File* et choisir *New Patcher*.



Un *patcher* est la page blanche de Max. C'est l'espace où on *patch* les objets entre eux. Il y a un ensemble d'objets dit *interface*, puis il y a tous les autres objets qui sont créés à l'intérieur de la boîte d'objet (« object box »). Le nouveau patcher s'ouvre en mode *edit* (*unlocked*, *ouvert*). Pour passer en mode *exécution* (*fermé*, ou *locked*), il faut faire « commande-E » ou « commande-click » dans un espace vide.

object palette

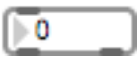
Pour voir la palette d'objet, et en choisir un à créer, il faut soit double-cliquer dans le patcher, soit taper « p ».



Vous pouvez vous balader à l'intérieur avec les flêches, et entre les catégories avec la touche « tabulation ». Aussi avec la souris, évidemment.

D'abord, aller sur la catégorie *All* ou *Basic* et choisir le sixième objet : *number*. Cliquer avec la souris ou taper « retour ».

Number est un objet *user interface* pour afficher et envoyer des valeurs numériques entières (*integer* ou *int*). Pour le manipuler, il faut passer le patcher en mode *exécution* (*fermé*, ou *unlocked*) avec « commande E ».

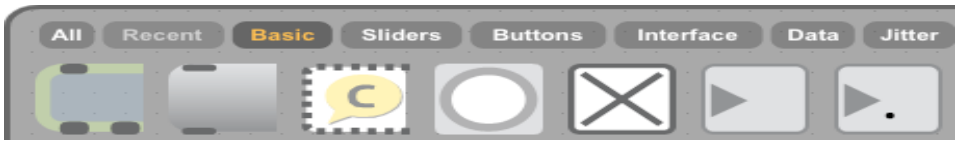


On peux aussi manipuler un object *user interface* tout en gardant le patcher en mode *Edit* en cliquant sur l'objet pendant qu'on appuie sur « commande ».

les *messages*

On va créer quelques objets et ainsi explorer un concept absolument essentiel dans Max, les *messages*.

Avec le ***number*** déjà crée, rajouter un ***flonum*** (*float number*), un ***toggle***, un ***button***, un ***comment*** et un ***message*** (boîte à messages).



message *comment* *button* *toggle* *number* *flonum*

On peut déplacer les objets avec la souris, et afficher des informations sur leurs entrées et sorties en plaçant la souris directement sur les points noirs d'entrée/sortie (sans cliquer).

int

Nombres entiers, positifs et négatifs (***number***, ***toggle***, ***int***)

float

Nombres avec décimal, positifs et négatifs (***flonum***, ***float***)

symbol

Une chaîne de caractères sans espaces, souvent dans une boîte à messages (***message***).S'il faut avoir des espaces dans une chaîne de caractères pour être reconnu comme un seul symbol, il faut mettre le tout entre guillemet : "chaîne de caractères avec espaces"

bang

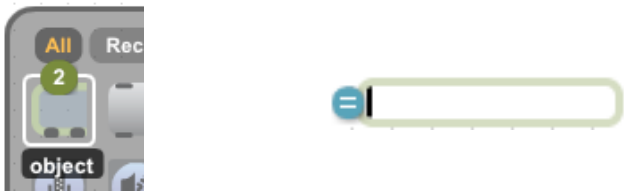
Un message pour déclencher des envoies ou des calculs d'autres objets. C'est un message qui veut dire « maintenant ! ». (***button***)

list

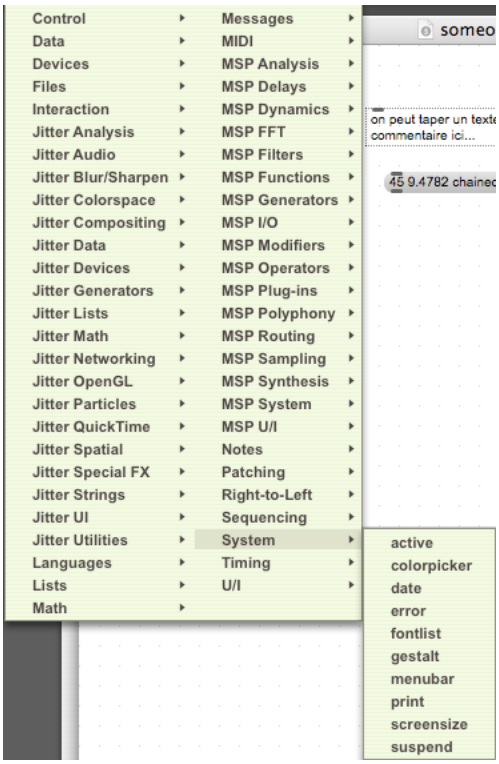
Une série des 4 types de messages précédents séparés par des espaces (c'est pour ça que les symbols ne peuvent pas contenir des espaces). Ils sont souvent dans des boîtes à messages aussi, mais il y a plusieurs objets qui traitent les données en forme de *liste*.

l'objet *object*

Choisir le premier objet de la palette d'objet – ***object*** (on peux aussi taper « n » sur le clavier pour créer un « n »ouvel objet).



Placer la souris sur le bord gauche de cet objet vide pour faire afficher un signe « = ». En cliquant dessus, vous ouvrez la liste des objets :



Cliquer sur *System->**print***, et un nouvel objet ***print*** apparaît dans le patch avec le curseur qui clignote à l'intérieur.

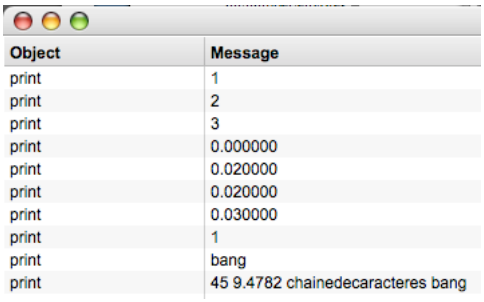
Cliquer en dehors de l'objet ou taper sur la touche « enter » sur le clavier pour *instancier* l'objet (le charger dans le patch comme une « instance » de l'objet).

Maintenant connecter les objets ***int number***, ***float number***, ***toggle***, ***button*** et ***message*** au ***print*** en positionnant la souris sur la sortie d'un objet…

puis en cliquant, glissant, puis relâchant sur l'entrée du ***print***.

Une fois les objets connectés, passer le patcher en mode « fermé » (« locked », « mode exécution ») en appuyant sur « command-E » sur le clavier, ou bien en appuyant sur la touche « command » puis cliquant la souris dans un espace blanc du patch.

Maintenant on peux cliquer sur les objets envoyer leurs contenus ou modifier leurs valeurs en glissant la souris (***int*** et ***float***) Dès qu'on clique sur un objet ou on modifie son contenu, les messages sortent et passent dans le ***print*** ce qui les fait apparaître dans la fenêtre Max



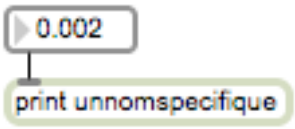
Object	Message
print	1
print	2
print	3
print	0.000000
print	0.020000
print	0.020000
print	0.030000
print	1
print	bang
print	45 9.4782 chainedecaracteres bang

On voit bien la différence entre les types de messages, et que le ***toggle*** est tout simplement un ***int*** limité à envoyer uniquement « 0 » et « 1 », etc…

les arguments des objets

Si on a plusieurs objets ***print*** dans un patch, on peux faciliter le repérage des indices en mettant un *argument* dans l'objet – un message qui précise des caractéristiques d'un objet, écrit dans l'objet après le nom + un espace »

Prenez un nouvel objet ***print*** mais cette fois-ci rajouter « espace » puis un nom (sans espaces !) :



Maintenant quand on modifie une valeur envoyée dans de ***print*** le nom de notre print précède la valeur dans la fenêtre Max.

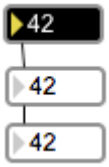
unnomspecifique	0.000200
unnomspecifique	0.000400
unnomspecifique	0.000800
unnomspecifique	0.001100
unnomspecifique	0.001200

Le plupart des objets Max peuvent avoir des arguments. Leurs significations sont spécifiques à chaque objet. On les verra petit à petit.

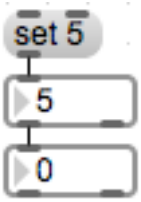
le message *set* et les variables dans les messages

Il y a un message spécial compris par beaucoup d'objets qui est *set*. Avec le message *set* on peux changer le contenu d'un objet sans déclencher une sortie de sa valeur.

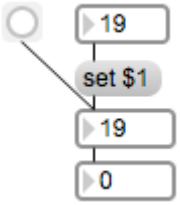
Normalement, si un objet reçoit une valeur dans son entrée gauche, il va déclencher une sortie du résultat de cette valeur. Pour un ***number*** ça veux dire de repasser la valeur à sa sortie.



Si on inclue une valeur après un message *set* dans une boîte à message, on peux faire placer une valeur sans qu'il y ait une sortie. Il faut cliquer sur le message en mode exécution :



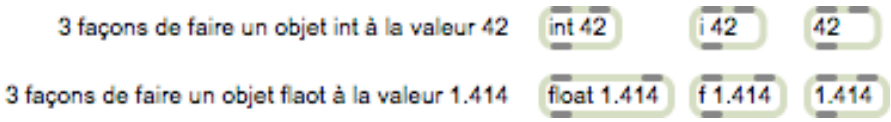
Et si on met un « $1 » dans un message, ça fonction comme variable, étant remplacé par une valeur qui rentre dans le message. Un button (*bang*) dans un objet fait sortir la valeur (ou calculer le résultat, selon la nature de l'objet).



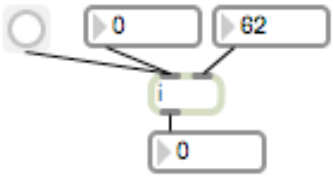
La valeur « 19 » se met dans le 2e number mais ne passe pas au 3e. En cliquant sur le « bang » du 2e, la valeur passe au 3e.

Les objets *int* et *float*

Pour les ints et les floats il y a aussi des objets ***int*** et ***float***. prenez un nouvel objet et taper « int » out simplement « i » pour créer un objet ***int***. Pareil pour un float en tapant « float » ou « f ». Par défaut ces objets ont comme valeur interne « 0 ». Si on veux autre chose comme valeur de départ il faut le taper comme un argument. Comme raccourci dans les deux cas, on peux aussi tout simplement taper la valeur sans le non de l'objet, et ça crée un objet du même type que la valeur :

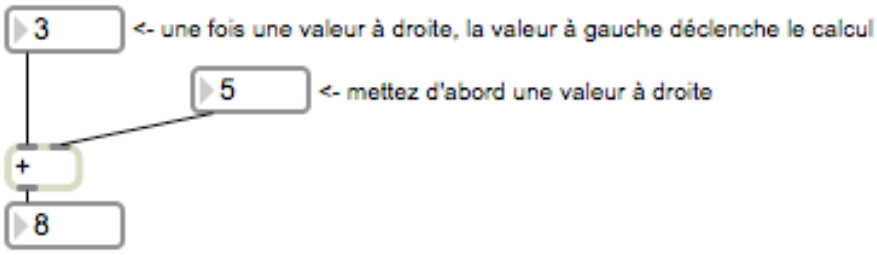


Si on fait rentrer une valeur par l'entrée gauche, la valeur déclenche la sortie de l'objet. Si on veux faire rentrer une valeur sans déclencher une sortie, il faut simplement passer la valeur dans l'entrée droite. Un « bang » ou une valeur à gauche fait sortir la valeur.



Un objet typique : « *+* »

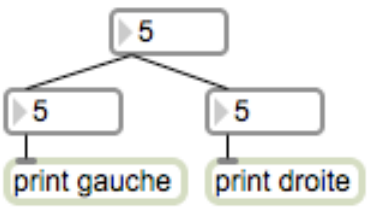
Prenez un nouvel objet, catégorie : Math, objet : « ***+*** ». Puis connecter trois nouveaux ***number***, un dans chaque entrée de « ***+*** » et un à sa sortie.



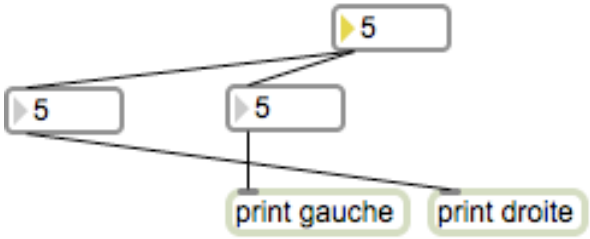
Une fois en mode « locked », mettez une valeur dans le ***number*** à droite en premier. Le « ***+*** » ne fait pas de calcul. Mettez maintenant une valeur dans le ***number*** gauche. Le calcul du « ***+*** » est déclenché.

Priorité droite / gauche

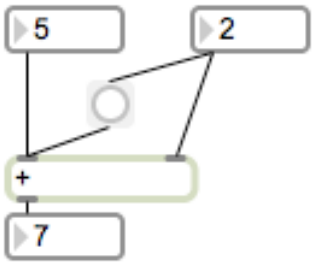
Les entrées et les sorties des objets dans Max sont réglées par une priorité générale de **DROITE / GAUCHE**. Que ça soit l'ordre par lequel il faut envoyer les messages dans un objet, ou l'ordre des sorties d'un objets avec plusieurs sorties. Ça s'applique même à l'ordre des envoies des messages selon l'emplacement des objets dans un patch. Un objet qui envoie à deux autres objets envoie d'abord à l'objet à droite puis à l'objet à gauche. Créez le patch suivant et regarder dans la fenêtre Max en bougeant le ***number*** en haut.



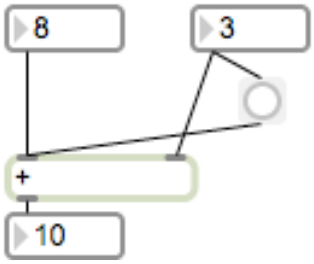
Si on déplace le ***number*** droite à gauche du ***number*** gauche, l'ordre est inversé.



Pour revenir à l'objet « *+* », si on veux changer la valeur gauche et déclencher un calcul, il faut rajouter un « bang » qui envoie à gauche juste APRES chaque nouvelle valeur à droite.



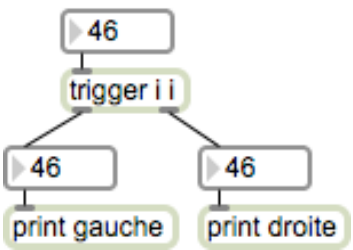
ATTENTION. Le ***bouton*** doit être placé à GAUCHE de l'entrée droite du « *+* », sinon le « bang » sera envoyé AVANT que la valeur n'arrive à droite du « *+* » !

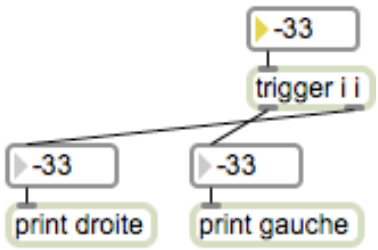


Ça donne donc des faux résultats car l'entrée des valeurs est décalée. Ici, le « 8 » est additionné à « 2 » qui est la valeur PRECEDENTE à droite.

L'objet *trigger*

Pour mieux maîtriser la question de la priorité, l'objet trigger permet de la préciser sans se soucier de la position des objets. Les arguments de trigger définissent le type de message. Ici « i » pour int. Si on change l'emplacement, l'ordre reste inchangé.





ANNEXE :

Object Defaults

On peux personnaliser les valeurs par défauts (couleurs, comportements, etc) de beaucoup d'objets. Ce n'est peut-être pas essentiel au début, mais c'est bon à savoir. Aller dans le menu *Options* et choisir *Object Defaults*.