

**Max/MSP Introducion ::
création audio et visuelle en temps réel
Cours 3 : introduction vidéo**

**ENSEIGNANT:
Alexandre Quessy**

2008/2009

Société des arts technologiques
1197 boulevard Saint-Laurent
C.P. 1083 Succursale Desjardins
Montréal (Québec) H5B 1C2

DESCRIPTION DU COURS

Descriptif :

Une formation de 18 heures en trois cours animée par Alexandre Quessy. Les cours permettent de s'introduire à l'utilisation de Max/MSP, un logiciel de création audio et visuelle en temps réel. La formation complète est divisée en trois sessions de 6 heures : introduction générale, introduction à l'audio (la composante « MSP » du logiciel), introduction à la vidéo avec Jitter.

Contenu :

- Prendre connaissance des possibilités offertes par le dataflow;
- Comprendre le flot des données entre les objets dans Max/MSP;
- Apprendre à concevoir des programmes interactifs;
- Se familiariser avec les principaux objets de création audio et musicale;
- Se familiariser avec la création vidéo, visuelle et d'images générées;
- Expérimenter la capture vidéo et la capture sonore en direct;

Matériel requis :

Aucun matériel n'est requis. Les étudiants sont invités à apporter leur ordinateur avec le logiciel Max 5 installé (<http://www.cycling74.com>). Max 5 est compatible avec Mac OS X et Windows. Il est gratuit pour les 30 jours, et il est possible par la suite de l'acheter. Des forfaits spéciaux sont disponibles pour les étudiants : il suffit d'envoyer l'image de notre carte étudiante par courriel à la compagnie pour en bénéficier.

Cours 2 : Introduction vidéo

Objectifs :

- Se familiariser avec les notions de l'environnement de programmation Jitter qui permet un contrôle en temps réel de la vidéo et de la 3D;
- Comprendre aux notions de bases de la vidéo tel que les pixels, la couleur, les matrices, les effets visuels, la mémoire tampon et le rendu OpenGL.

Prérequis : Max/MSP Introduction

Durée : 6 heures

Théorie

Les bases de Max/MSP : objets et messages

Lorsqu'on utilise une patche, on peut cliquer sur les boîtes de message, les boutons et les chiffres. Quand on dévérrouille la patche, il est possible de modifier les connexions entre les objets et les autres éléments dans la patche. On peut également ajouter d'autres éléments. C'est ce qu'on peut appeler le mode d'édition.

Les boîtes de message ont comme couleur de fond le gris ou une autre couleur. Les objets, eux, ont toujours le fond blanc.

Les connexions peuvent communiquer des messages, du signal sonore ou des matrices Jitter. Les lignes qui communiquent du signal audio sont jaune et noir, et celles qui communiquent des matrices sont vert et noir. Les autres sont simplement plus minces.

Les entrées des objets sont situées sur le dessus de leur boîte rectangulaire. Leur sorties sont situées en-dessous. Les messages vont donc de haut en bas. Pour connecter deux objets ou boîtes de messages, il faut être en mode d'édition et cliquer sur la sortie d'un objet, puis déplacer la sortie au-dessus de l'entrée d'une autre objet en gardant le bouton de la sortie enfoncé. Lorsque l'on relâche le bouton de la souris, une nouvelle connexion est née !

Dans la boîte d'un objet, le premier mot est le nom de l'objet comme tel. Ce mot définit son comportement. Les mots suivants sont des arguments de création de cet objet et influencent également, dans une moindre mesure, son comportement. Certains de ces arguments sont des attributs dont on peut changer la valeur au moyen de messages spéciaux. On peut savoir comment utiliser un certain objet en consultant sa patche d'aide au moyen de l'aide contextuelle (un clic droit) et en choisissant d'ouvrir sa patche d'aide. Le menu « Help » de Max/MSP contient également des tonnes de ressources utiles. Le bouton « i » en bas de chaque patche permet d'ouvrir l'inspecteur, une fenêtre qui en dit long sur l'objet qui est sélectionné au moment d'appuyer sur ce bouton. On peut copier-coller un ou plusieurs objets d'une patch à l'autre, un peu comme dans un logiciel d'illustration vectorielle.

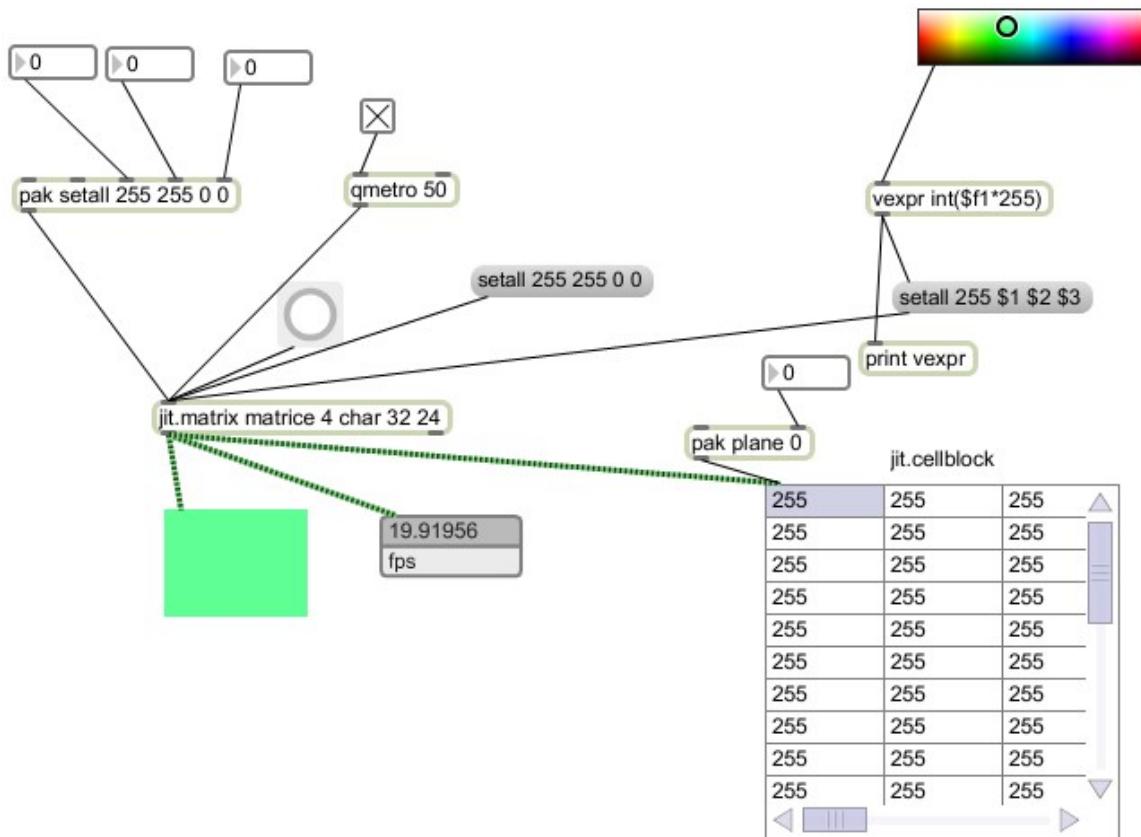
La plupart des objets qui gèrent des messages ont une seule entrée qui, lorsqu'elle reçoit un message, active l'envoi d'un autre message résultant d'un calcul par un de ses sorties. L'entrée active est celle à l'extrême gauche, et les autres ne sont pas actives. Elles ne font que changer les valeurs stockées à l'intérieur de l'objet. Tout les objets ne suivent pas ces critères. À la sortie, les messages sortent habituellement de la sortie la plus à droite en premier, et ensuite des autres. Si plusieurs objets sont connectés à une même sortie, c'est celui situé le plus à droite dans la patche qui reçoit le message en premier.

Les messages peuvent être des nombres entiers ou à virgule flottante, des « bang » ou encore des symboles. (du texte) Ils peuvent également être des listes comportant plusieurs atomes de l'un de ces types. Les objets de base pour gérer des listes de messages sont [pack] et [unpack]. Les messages comportant les symboles spéciaux « \$1 », « \$2 », etc. peuvent servir au même genre d'opération que [pack]. Les objets [select] et [route] servent à rediriger un message selon sa valeur ou celle de son premier atome. L'objet

[trigger] permet de séquencer et des convertir le type d'un message vers ses différentes sorties. Ses sorties sont activées de droite à gauche, tel que l'ordre standard des messages dans Max/MSP le stipule.

Jitter et les attributs des objets jit.*

Dans Max/MSP, on détermine le comportement et la valeur des objets au moyen de messages qu'on leur envoie, ou à des arguments donnés au moment de la création de ces objets. Les objets de la librairie Jitter ont poussé un peu plus loin ce système en utilisant le concept d'attribut. Les objets ont des attributs qui sont des variables contenant des chiffres, des mots ou le nom de ressources à utiliser. Lorsque l'on envoie un message commençant par le nom d'un des attributs d'un objet, on peut changer la valeur de celui-ci. On peut également donner la valeur initiale des attributs d'un objet en utilisant le caractère @ suivi du nom de l'attribut, puis de la valeur à lui donner.

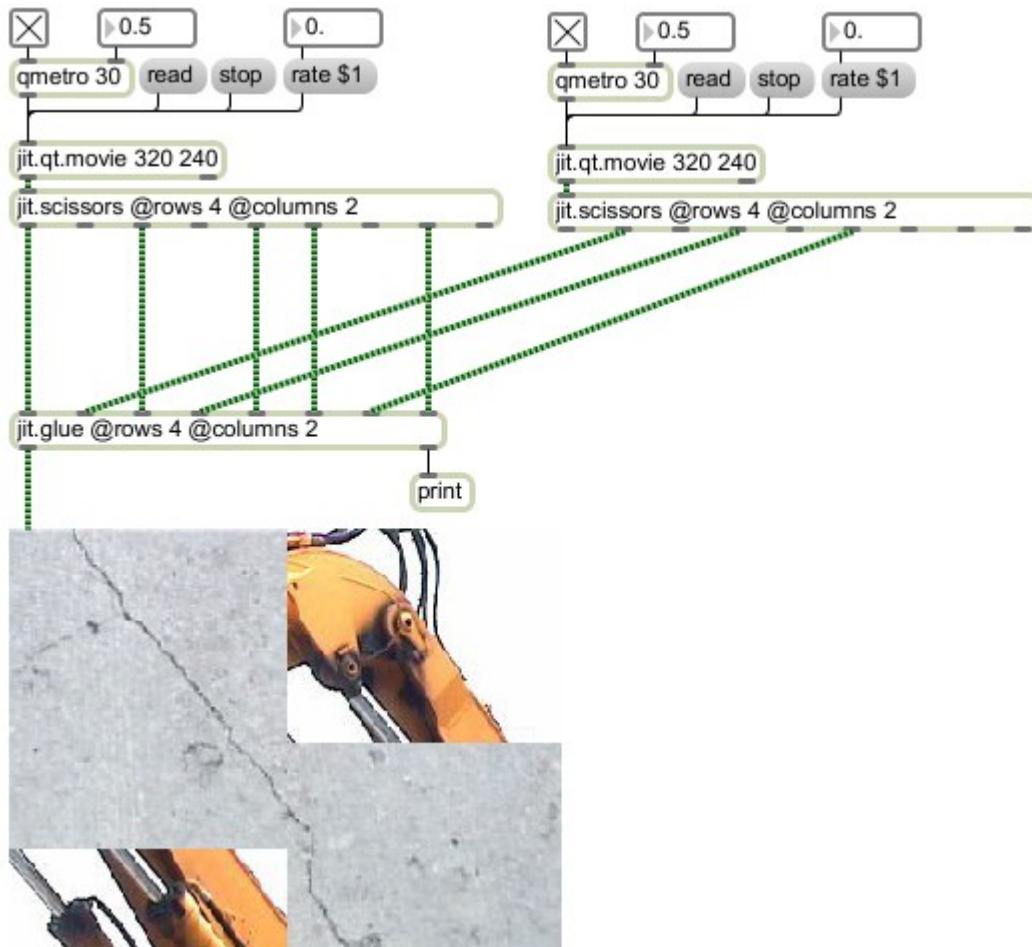


jit.matrix : une matrice de pixels

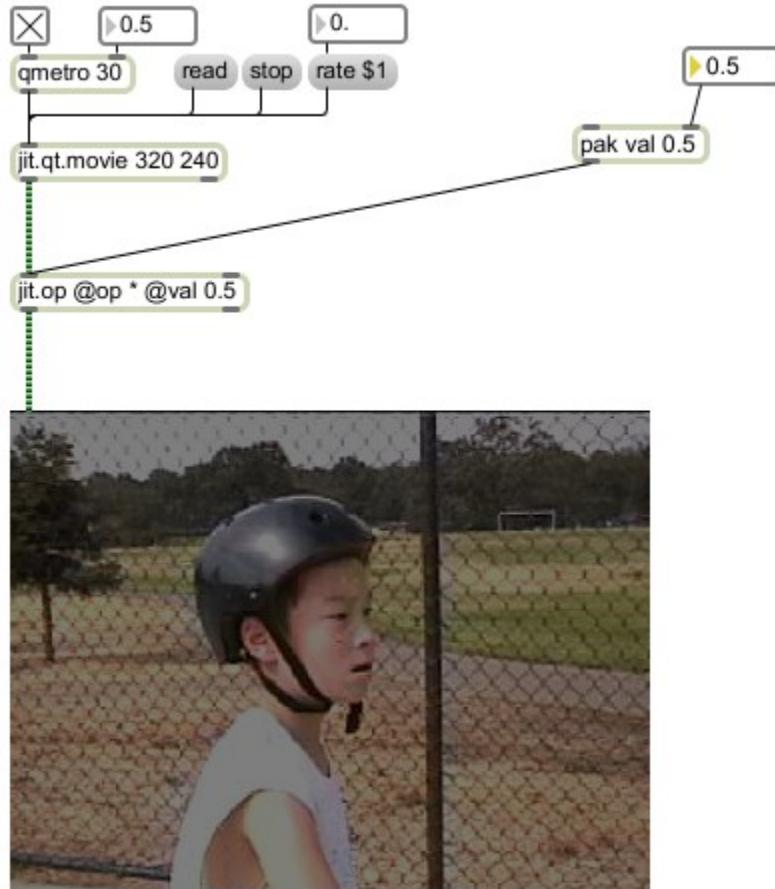
jit.matrix stocke une matrice de nombres. Elle peut avoir un nombre variable de dimensions. La dernière dimension est appelée "plan". Les matrices Jitter pour les images ont habituellement 4 plans : "ARGB" : la transparence, le rouge, le vert et le bleu. Les couleurs sont stockées sous la forme de nombres entiers entre 0 et 255. Chaque pixel est

donc une case comportant quatre plans.

On peut obtenir beaucoup d'informations sur une matrice avec les objets jit.matrixinfo, jit.cellblock, et jit.fpsgui.



Il est possible de combiner des matrices ensemble , de les découper, les recoller, etc.



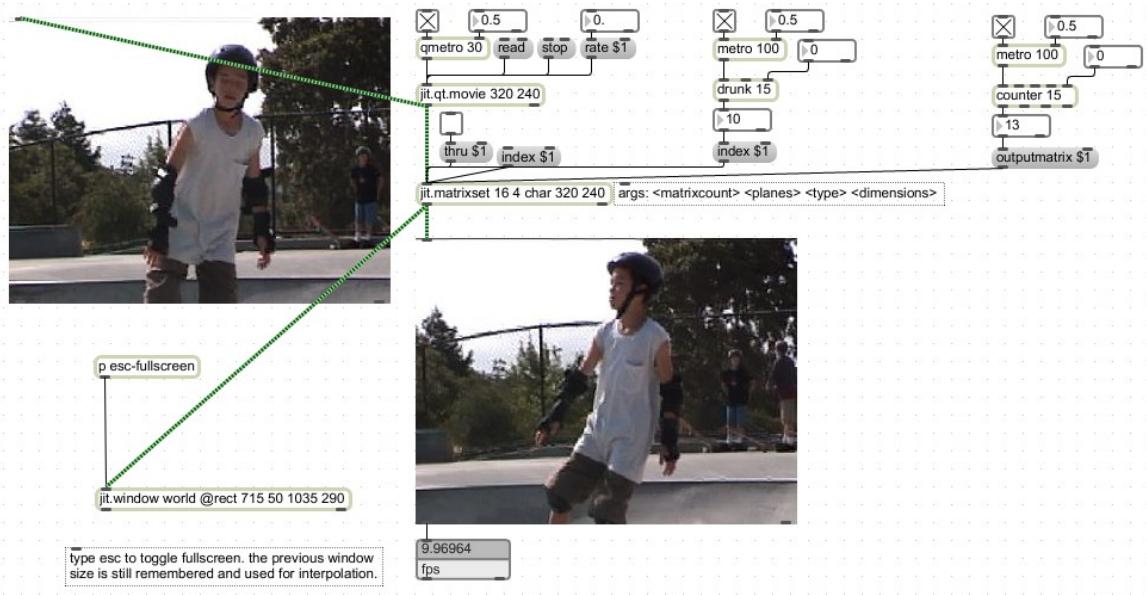
On peut aussi leur appliquer des opérations mathématiques.

jit.window : la fenêtre

jit.window est la fenêtre d'affichage. En l'entourant de quelques objets, (voir sa patche d'aide) on peut la contrôler avec la touche ESCAPE. jit.pwindow est une alternative à jit.window qui peut être très utile en phase de développement, car elle est placée à même la "patche".

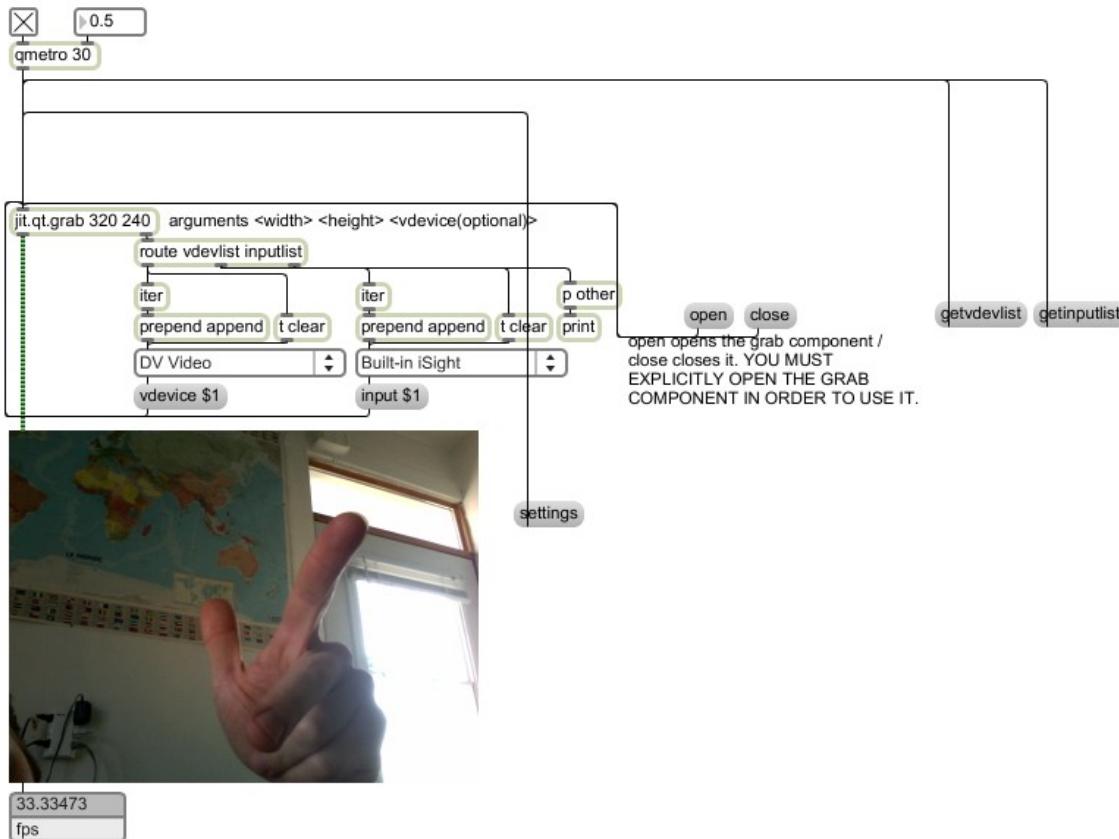
jit.qt.movie : lire des clips vidéo

jit.qt.movie sert à lire des fichiers quicktime, des images et d'autres types de documents médiatiques.



jit.matrixset : enregistrer des boucles vidéo

jit.matrixset sert à capturer des images et à les faire jouer à nouveau. Dans un "matrix set", on peut stocker une série de matrices. C'est un contenant pour un clip, une séquence de "frames".



jit.qt.grab : capturer les entrées caméra

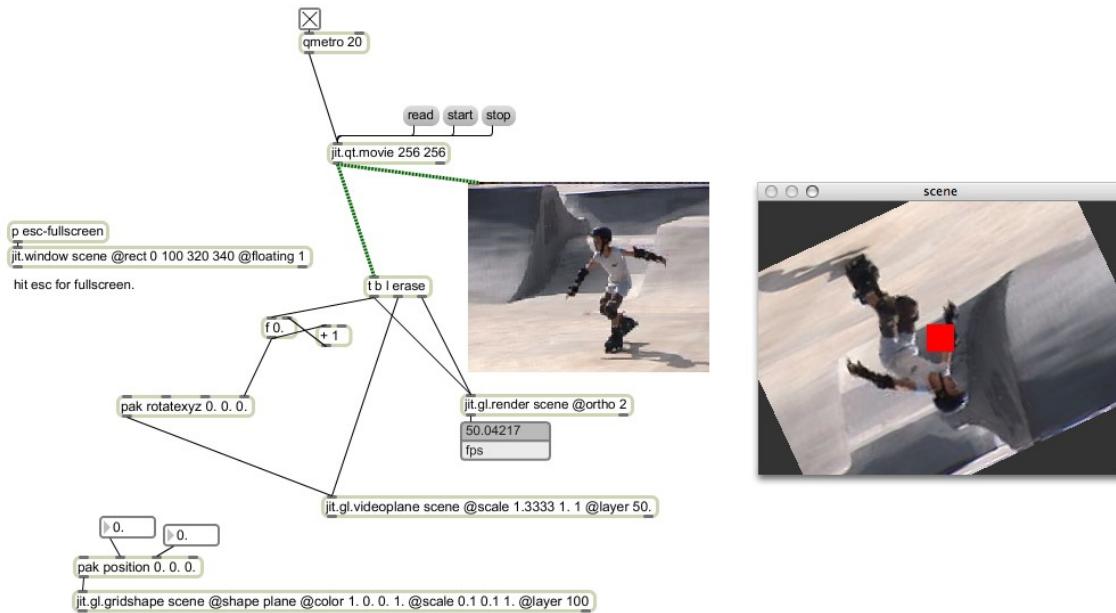
jit.qt.grab et jit.dx.grab sont les entrées caméra sur Mac et Windows, respectivement. Cet objet nous donne une matrice à chaque nouveau frame. On peut utiliser avec celui habituellement un signal provenant d'une caméra par une connection Firewire, USB, S-Video ou Composite. Il s'agit d'utiliser un pilote qui soit supporté par le système d'exploitation. Sur Mac OS X, il s'agit de Quicktime, le plus souvent. Sur Windows, c'est Direct-X qui est le pilote, avec l'objet jit.dx.grab.

Un signal de caméra peut être dans le format NTSC ou un autre. Si c'est le cas, le "frame rate" serait de 29.97 FPS. Il serait approprié de lire un image à chaque fois qu'il en a une nouvelle, et donc avec un qmetro à 33.36 ms d'intervalle.

Il ne faut pas oublier de lui envoyer le message "open" si on veut obtenir une image vidéo. L'objet graphique umenu est un petit menu déroulant qui peut faciliter le choix d'une caméra pour l'objet jit.qt.grab.

qmetro : le métronome à utiliser avec Jitter

Pourquoi un qmetro plutôt qu'un métro ? Le qmetro a comme un fusible qui l'empêche de ralentir tout le système s'il y a plus de tâches à faire que de temps. Il va sauter des "frames". Ce "jump cut" est perceptible, et même désagréable lorsque la scène rendu comporte des animations. (déplacement, rotation, juxtaposition et couleur)



jit.gl.render : le rendu d'une scène OpenGL

jit.gl.render sert à faire le rendu d'une scène OpenGL.

OpenGL est un standard de l'industrie des cartes vidéo pour ordinateurs et jeux vidéo. Il s'agit d'un ensemble de fonctions pour contrôler la forme, les couleurs et les images pour le rendu d'une scène en trois dimensions. Les scènes peuvent également être en deux dimensions dans le cas de l'imagerie numérique à partir de séquences vidéo.

L'avantage d'OpenGL est que ce type de rendu utilise les ressources matérielles de la carte vidéo. (Nvidia ou ATI, par exemple) Ceci libère le CPU (Microprocesseur central) pour lui permettre d'effectuer d'autres tâches durant ce temps. En effet, le CPU est déjà très occupé avec la lecture de flux vidéo provenant du disque dur ou d'une caméra vidéo. Le rendu peut être délégué à la carte graphique.

Les images sont transférées à la carte graphique et y sont stockées sous la forme de textures. Ces textures peuvent être appliquées sur des formes. La carte graphique applique ces images à des formes 3D virtuelles. Comme dans un jeu vidéo. Il peut y avoir des sources de lumière, et des formes étranges ou réalistes.

Notez que le signal VGA ou DVI qui va de l'ordinateur à un projecteur numérique a souvent un taux de 60 Hz. Il en résulte que 60 image par second est une bonne vitesse pour un rendu OpenGL.

Dans Jitter, tous les objets OpenGL sont associés à un contexte de rendu. On peut les reconnaître facilement, car leur nom commence par `jit.gl.*`. Le premier argument qu'on leur donne est le nom d'un contexte de rendu sur lequel les dessiner. Ce mot est au choix du programmeur. La plupart des projets OpenGL avec Jitter ont un seul contexte de rendu. C'est le premier argument des objets comme `jit.gl.render` et `jit.gl.gridshape`. Un contexte

de rendu, c'est une scène OpenGL.

On peut déterminer l'ordre dans lequel les objets jit.gl.* sont dessinés grâce à leur attribut "layer". Ils sont dessinés du plus petit au plus grand. L'objet à dessiner en dernier devrait avoir l'attribut "layer" le plus grand d'un contexte.

jit.gl.videoplane : une forme avec de la vidéo

Dans une chaîne de traitement typique OpenGL, les textures sont appliquées sur les formes 3D selon la couleur de ces formes, et de la lumière dans le monde tridimensionnel, si celle-ci est activée. On peut ainsi superposer plusieurs images avec une transparence (alpha) variable. Toutes les couleurs sont en RGBA entre zéro et un. De 0.0 à 1.0.

Les images qu'on applique sur un jit.gl.videoplane sont des matrices. Chaque nouvelle matrice correspond à un nouveau "frame".

Les messages "position" et "rotatexyz"

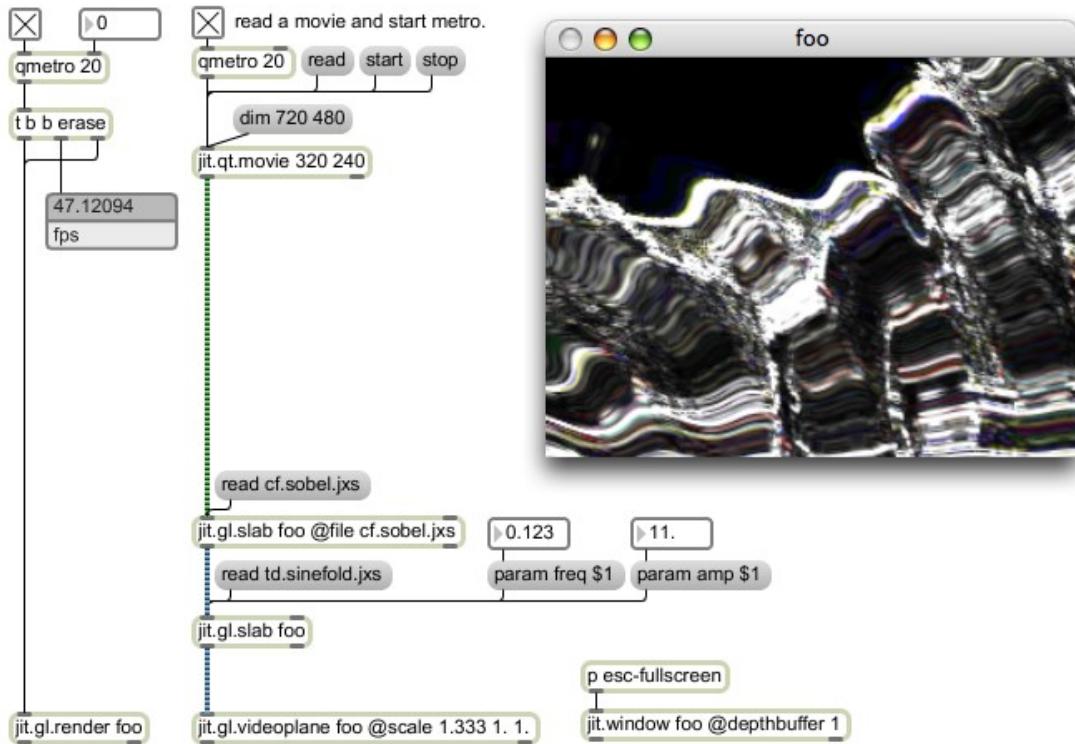
Les objets jit.gl.* peuvent être déplacés dans l'espace 3D grâce à des messages. Le mot-clé peut être translatexyz ou rotatexyz, par exemple.

Le secret pour générer des animations fluides est de baser l'interpolation sur le temps, et non sur le numéro de "frame". Ainsi, lorsque le qmetro saute un "frame", ce n'est pas trop facilement décelable à l'oeil. Ceci n'est valide que dans le cas des animation 2D et 3D, et non de la simple vidéo.

jit.gl.gridshape : différentes formes à dessiner

Avec OpenGL, on peut dessiner des formes 2D ou 3D. L'object jit.gl.gridshape permet de dessiner à l'écran plusieurs sortes de formes. Cercle, carré ou cube : il les plusieurs formes simples. On peut même y appliquer une jit.gl.texture provenant d'une caméra vidéo ou d'une clip. Notez que dans le cas de l'objet jit.gl.videoplane, il n'est pas nécessaire d'utiliser une jit.gl.texture avant, alors que dans ce cas-ci, oui.

Il existe plusieurs autres objets pour dessiner dans un contexte OpenGL, comme par exemple l'objet jit.gl.text3d qui dessine du texte.



jit.gl.shader : les shaders

Les créateurs de jeux vidéo souhaitent synthétiser des ombres dans leurs jeux. Or, un rendu OpenGL tient compte automatiquement des sources de lumière, mais pas de l'ombre qu'un objet pourrait porter sur un autre. Pour pallier à cet handicap, des programmeurs ont pensé à un moyen d'écrire des scripts pour modifier le rendu normal des couleurs en OpenGL. Ces scripts sont appelés des "shaders". Il s'agit de textes donnant des instructions à la carte graphique. De nombreux scripts sont disponibles avec Jitter ou sur Internet.

On peut chaîner plusieurs effets shaders sur une même texture. On peut même combiner plusieurs textures selon des formules conditionnelles. Un exemple familier serait l'écran bleu de la météo à la télé. Il faut faire un calcul, et vite, parce que c'est en direct. On fait appel à un shader sur la carte graphique plutôt qu'à des opérations sur les matrices, qui seraient coûteuses en temps d'exécution sur le processeur central de l'ordinateur.

RÉFÉRENCES ET LIENS WEB:

Les tutoriels de la section Jitter de l'aide de Max/MSP sont la source préférée d'informations quant à l'utilisation des objets Jitter. Le site web de la compagnie Cycling '74 est également riche en informations récentes.

The Max Objects Database

Des patches et des externals

<http://www.maxobjects.com/>

La page de Max/MSP 5 site de Cycling 74

Là où tout commence...

<http://cycling74.com/products/max5>

Les forums de la compagnie

Parmi ceux-ci, celui de MaxMSP et celui de Jitter nous intéressent.

<http://cycling74.com/forums/index.php>

La documentation en ligne de Max 5

Les tutoriels vidéo sont excellents. Les tables des matières des « MSP Tutorials » et des « Jitter Tutorials » se trouvent à la droite de l'écran.

<http://www.cycling74.com/docs/max5/vignettes/intro/docintro.html>