

38-Configuration de base des performances

Dans ce chapitre, nous allons parcourir les différentes étapes de la configuration de votre patch Jitter pour une performance en direct. En cours de route, nous verrons comment utiliser un second moniteur vidéo pour une sortie plein écran.

- Ouvrez le patch du didacticiel.

La plupart des patches que nous avons examinés jusqu'à présent utilisent l'objet *jit.pwindow* pour afficher les données vidéo. Bien que cet objet soit extrêmement pratique pour regarder une vidéo à l'intérieur de votre patch, il n'est pas idéal pour les performances en direct (sauf si vous voulez que le public voie aussi votre patch). Dans ce cas, nous utilisons l'objet *jit.window* (dont vous vous souvenez peut-être du didacticiel 1 Lire un film).

- Cliquez sur la boîte de *message* intitulée *read dishes.mov* et activez le *toggle* intitulé Start Movie pour commencer la lecture du film. Vous devriez voir le film en cours de lecture dans la fenêtre de l'objet *jit.window*.

Quel est cet attribut **@noaccel 1** tapé dans l'objet *jit.window*? Il indique à l'objet que nous voulons désactiver l'accélération OpenGL. Pour les besoins de ce didacticiel, nous allons ignorer certaines des fonctionnalités les plus avancées de *jit.window* - en particulier la possibilité d'utiliser le support OpenGL de Jitter pour améliorer l'apparence et la vitesse globale des données vidéo. Mais ne vous inquiétez pas. Ces fonctionnalités seront abordées dans les chapitres ultérieurs du didacticiel.

Jetons un coup d'œil à certaines des capacités de l'objet *jit.window*. En plus de déplacer et de dimensionner la fenêtre à l'aide de la souris (un clic dans le coin inférieur droit nous permet de redimensionner la fenêtre manuellement), nous pouvons envoyer des messages à l'objet et provoquer le déplacement ou le redimensionnement de la fenêtre, nous pouvons supprimer la bordure de la fenêtre, faire en sorte que la fenêtre flotte au-dessus de toutes les autres fenêtres, et faire en sorte que la fenêtre remplisse tout l'écran.

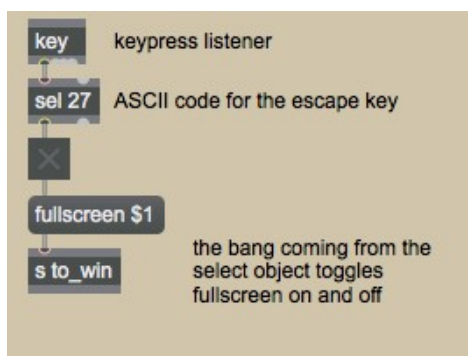


Redimensionner la fenêtre de l'objet *jit.window* avec la souris.

- Cliquez sur la boîte de *message* étiquetée **size 160 120** et voyez ce qui se passe. Tout comme l'objet *jit.pwindow*, l'objet *jit.window* met automatiquement à l'échelle la vidéo entrante pour qu'elle s'adapte à la taille de la fenêtre lorsqu'il reçoit le message **size 160 120**. Cliquez sur la boîte de *message* intitulée **size 320 240** pour revenir à la taille d'origine.
- Cliquez sur la boîte de *message* intitulée **pos 300 400** et voyez ce qui se passe. Nous pouvons déplacer la fenêtre vers n'importe quelle position valide de l'écran, simplement en lui envoyant le message **pos** (position) suivi des coordonnées. Nous allons utiliser cette fonction dans un instant pour créer un patch de positionnement automatique de la fenêtre. Cliquez sur la boîte de message intitulée **pos 800 100** pour revenir à la position d'origine.
- Désactivez le *toggle* intitulé **Border On / Off** et voyez ce qui se passe. La barre de titre de la fenêtre et la bordure (sous Windows) de la fenêtre devraient avoir disparu. Activez le *toggle* pour faire revenir la bordure.
- Activez le *toggle* **Float On / Off** et voyez ce qui se passe. Une fenêtre flottante "flotte" au-dessus de toutes les fenêtres non flottantes et ne peut jamais être envoyée à l'arrière (bien qu'une autre fenêtre flottante puisse la recouvrir). Faire flotter une fenêtre peut vous aider à vous assurer que vos données vidéo sont toujours visibles si vous avez un patch complexe avec de nombreuses fenêtres. Désactivez le *toggle* pour désactiver le flottement pour cette fenêtre.
- Enfin, appuyez sur la touche Echap (**esc**) du clavier de votre ordinateur et observez comment la fenêtre remplit automatiquement tout l'écran de votre moniteur principal. Appuyez à nouveau sur la touche Echap pour réduire la fenêtre à sa taille précédente.

Si vous regardez dans le patch, nous pouvons voir clairement comment fonctionne cette dernière astuce. Nous avons simplement utilisé l'objet Max *key* en combinaison avec un objet *select* pour employer la touche *esc* (code ASCII 27). Lorsque cette touche est enfoncée, un **bang** est envoyé de l'objet *select* au *toggle*, ce qui fait passer son état de 1 à 0 ou de 0 à 1 (selon ce qu'il était avant de recevoir le **bang**). Cette valeur est envoyée à la boîte de *message* intitulée **fullscreen \$1**, provoquant l'envoi des messages **fullscreen 1** et **fullscreen 0** à l'objet *jit.window*.

Relier ce message à une touche est particulièrement astucieux, car une fois que la fenêtre est en plein écran, il est parfois un peu difficile de revenir à votre patch! Soit dit en passant, l'objet *jit.window* possède un attribut utile - **fsmenubar** - qui lui indique s'il doit ou non masquer la barre de menu du système lorsque l'objet est en mode plein écran. Par défaut, l'attribut **fsmenubar** est défini sur 1, ce qui signifie que la barre de menus doit rester où elle se trouve.

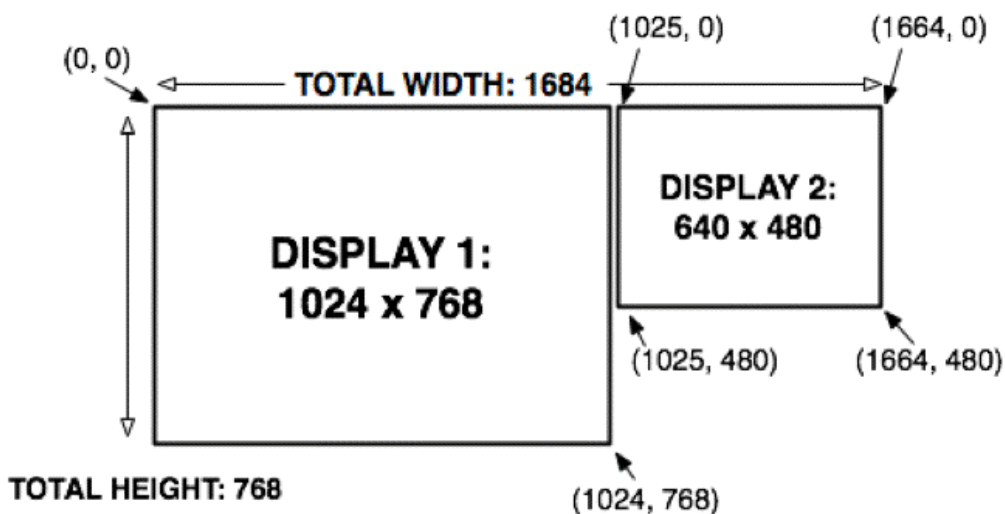


Basculer l'état plein écran de l'objet *jit.window* avec la touche d'échappement.

C'est recouvert

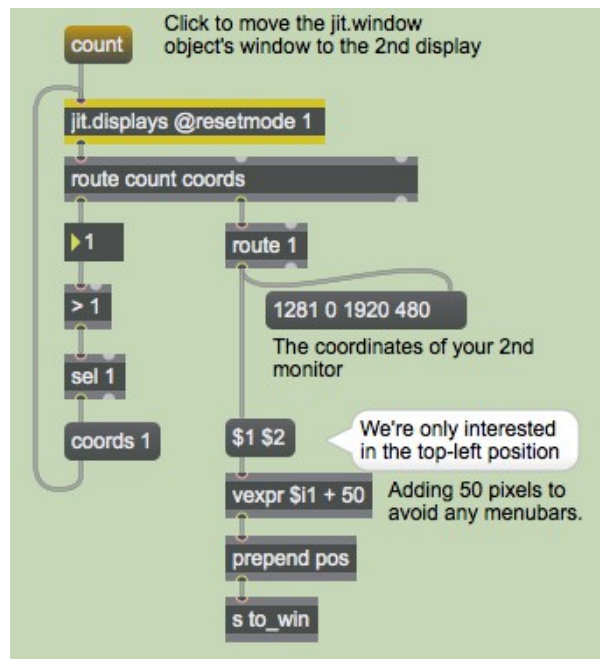
La plupart du temps, cependant, nous voulons être en mesure d'utiliser notre patch sur un écran, tout en regardant la sortie sur un autre. L'objet *jit.window* est capable de placer une fenêtre sur n'importe quel écran connecté à votre ordinateur (ou un dispositif d'affichage, comme un vidéoprojecteur).

En ce qui concerne votre ordinateur, une fois que vous avez connecté des écrans supplémentaires dans une configuration d'extension (c'est-à-dire que les écrans ne se reflètent pas l'un l'autre), vous avez simplement une grande zone d'affichage. Pour déplacer des fenêtres vers un deuxième écran, il suffit de connaître les bonnes coordonnées.



Vue d'un système à deux moniteurs, avec coordonnées.

Bien sûr, déterminer où se trouve votre deuxième moniteur et faire le calcul est un peu ennuyeux. Heureusement, Jitter inclut un objet, *jit.displays*, qui peut calculer tout cela pour vous.



L'objet *jit.displays* en action.

L'utilisation de l'objet *jit.displays* pour énumérer le nombre et les coordonnées des moniteurs connectés est simple. Le message **count** fait en sorte que le mot **count**, suivi du nombre d'écrans connectés, soit envoyé par la sortie gauche de l'objet. Dans notre patch, nous testons pour nous assurer qu'il y a plus d'un écran attaché utilisant l'objet **>** (plus grand que). Si tel est le cas, le message **coords 1** est envoyé à l'objet *jit.displays*, demandant les coordonnées du moniteur 1 (les moniteurs sont numérotés à partir de 0 - votre écran principal est généralement le moniteur 0).

- Cliquez sur la boîte de *message* intitulée **count**. Si vous avez un deuxième moniteur connecté à votre ordinateur, la fenêtre de l'objet *jit.window* devrait y apparaître. En utilisant simplement les coordonnées haut-gauche du deuxième écran comme base d'un calcul simple (nous avons ajouté 50 pixels à chaque coordonnée pour être sûrs d'éviter quelques barres de menus), nous avons pu envoyer un message **pos** à l'objet *jit.window* et positionner la fenêtre carrément sur celui-ci.
- Appuyez à nouveau sur la touche Echap. La fenêtre devrait passer en mode plein écran, mais cette fois, sur le deuxième écran. L'objet *jit.window* détermine l'écran sur lequel il se trouve avant de passer en mode plein écran, afin de s'assurer que la zone correcte de l'écran est couverte.
- Appuyez à nouveau sur la touche Echap pour désactiver le mode plein écran.

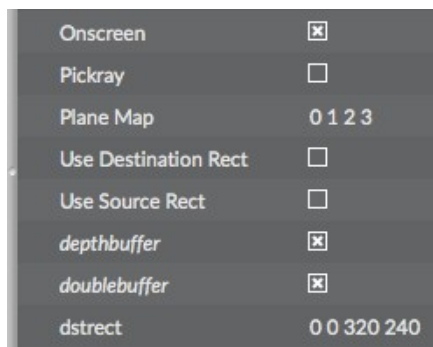
Bien entendu, l'objet *jit.displays* fonctionne avec des systèmes comprenant plus de deux écrans.

En règle générale, vous voudrez conserver une variante de cette partie du patch dans le cadre de votre configuration de performance. Nous avons remarqué que les paramètres de position d'un deuxième (ou troisième) écran ne restent pas toujours les mêmes d'une utilisation à l'autre, lorsque nous connectons différents types de périphériques à nos ordinateurs. L'utilisation de l'objet *jit.displays* pour déterminer automatiquement la position exacte de votre matériel connecté et déplacer votre fenêtre de sortie élimine une partie des tracas et des soucis liés à l'installation dans de nouveaux lieux ou avec du matériel inconnu.

Un petit conseil sur les performances

Maintenant que nous maîtrisons l'objet *jit.window*, tournons notre attention vers l'objet *jit.pwindow*. Bien que le but de ce tutoriel ne soit pas d'offrir des trucs et astuces pour améliorer les performances de Jitter, l'objet *jit.pwindow* est si souvent utilisé comme un petit moniteur dans les contextes de performances live, qu'un regard rapide sur son effet sur la vitesse de lecture vaut le coup.

- Cliquez sur la cellule de l'objet *matrixctrl* intitulée ***jit.pwindow*** pour acheminer les données vidéo vers l'objet *jit.pwindow*. Vous devriez voir la vidéo à la fois dans la fenêtre de l'objet *jit.window* et dans l'objet *jit.pwindow*. Comparez les fréquences d'images (telles que rapportés par l'objet *jit.fpsgui*) avec l'objet *jit.pwindow* activé et désactivé. Sur notre ordinateur, nous obtenons environ 300 images par seconde lorsque l'objet *jit.pwindow* est désactivé, et environ 110 images par seconde lorsqu'il est activé. Avec l'objet *jit.pwindow* activé et l'objet *jit.window* désactivé, nous obtenons environ 260 images par seconde. Vos résultats varieront considérablement en fonction du système d'exploitation et du matériel de votre ordinateur.
- Déverrouillez le patch et ouvrez l'inspecteur de l'objet *jit.pwindow*.



Inspecteur de l'objet *jit.pwindow*

Voyez-vous la case à cocher intitulée *Onscreen* dans l'inspecteur de l'objet *jit.pwindow*? Décochez-la et comparez les fréquences d'images avec les deux fenêtres actives. Sur notre système, nous obtenons environ 160 images par seconde, soit une augmentation de près de 50%! *Chaque fois que vous utilisez un objet jit.pwindow qui est plus petit que la vidéo affichée* (ici, notre vidéo est de 320x240 et notre *jit.pwindow* est de 80x60), *vous devez désactiver le mode onscreen*. C'est une règle simple qui vous permettra d'obtenir la meilleure vitesse possible de votre patch de performance.

Les modes *onscreen* et *offscreen* utilisent algorithmes de dessin différents pour effectuer leur travail: les données sont dessinées directement sur l'écran en mode *onscreen*, tandis que le mode *offscreen* dessine les données d'abord dans un tampon hors écran, puis les copie sur l'écran. L'algorithme utilisé dans le mode *onscreen* ne favorise pas le sous-échantillonnage, de sorte que les performances ne sont pas aussi bonnes.

Le paramètre *Use onscreen* dans l'inspecteur de l'objet *jit.pwindow* est enregistré avec le patch, il ne doit donc être défini qu'une seule fois. Vous pouvez également définir l'attribut *onscreen* directement à partir de Max en utilisant des messages.

Une autre stratégie pour réduire l'impact des objets *jit.pwindow* dans votre patch de performances est de réduire la quantité de données dessinées. Les objets Max *speedlim* et *qlim* sont parfaits à cette fin, bien que nous n'allons pas en montrer l'utilisation dans ce tutoriel (l'objet *speedlim* est discuté dans le *tutoriel MIDI 3*).

Résumé

Dans ce didacticiel, nous avons appris quelques stratégies pour surmonter les défis typiques dans un environnement de performances: les paramètres d'apparence pour l'objet *jit.window*; le mode plein écran; l'utilisation de l'objet *jit.displays* pour détecter le matériel supplémentaire d'affichage connecté à votre ordinateur et auto configurer la position de la fenêtre d'un objet *jit.window*; et la gestion du mode de dessin à l'écran de l'objet *jit.pwindow* pour améliorer les performances de vos écrans intégrés.