

28-Visualisation de données

introduction

Ce tutoriel examinera différentes techniques pour créer et afficher des flux de données numériques dans l'environnement Max. Nous nous pencherons sur l'objet *multislider*, qui possède des modes permettant à la fois de *créer* et de *surveiller* des valeurs de données. Nous nous intéresserons aux Monitor **Watchpoints**, un outil de débogage que vous pouvez utiliser pour afficher les messages qui transitent par les cordons au fur et à mesure de leur génération.

Lorsque vous travaillez avec un système modulaire tel que Max, il est parfois difficile de garder une trace de tous les messages envoyés entre les objets. L'utilisation des outils disponibles pour surveiller les messages et les valeurs peut vous aider à affiner un patch ou à détecter les problèmes de fonctionnement d'un patch. Les deux outils que nous allons utiliser - les objets *multislider* et les watchpoints - fournissent deux mécanismes différents pour le suivi des données: le *multislider* est une bonne option pour le suivi de l'exécution (performance), tandis que les watchpoints sont précieux pendant la programmation et le débogage.

Pour ouvrir le patch du didacticiel, cliquez sur le bouton vert **Ouvrir didacticiel** dans le coin supérieur droit de la fenêtre de documentation.

Multislider comme générateur de données

Jetez un coup d'œil à notre tutoriel. Ce patch contient de nombreux objets qui affichent et génèrent différents types de graphiques - en fait, ce sont toutes des versions du même objet, un *multislider*. Nous commencerons par le patch étiqueté **1**, qui contient un *counter* qui va parcourir les différents curseurs du *multislider* et produire une note pour chaque étape en utilisant MIDI. La sortie du *multislider* est comprise entre **0** et **127** (la gamme de notes MIDI standard) et fournit **16** étapes de données séquencées. Si vous activez le *metro* avec le *toggle*, vous verrez le numéro d'étape généré par l'objet *counter*, ajouté au message **fetch**, et utilisé pour interroger les seize curseurs du *multislider*. C'est la base de la plupart des patches de séquençage par pas, avec soit un contenu généré automatiquement (comme le fait la section du patch basée sur les *uzi-et-drunk*), soit en manipulant directement les différents curseurs de l'objet.

La sortie de droite du *multislider* est utilisée pour cette application. Lorsqu'elle est utilisée comme une banque de curseurs (ce qui signifie que l'attribut **Slider Style** est défini sur **Thin Line** ou **Bar**), la sortie *gauche* envoie toutes les valeurs des curseurs sous la forme d'une liste unique, et uniquement lorsqu'un message **bang** ou un nouvel ensemble de données complet est reçu. Le message **fetch** envoie la valeur d'un curseur individuel (numéroté à partir de **1**) à la sortie droite.

Examinez l'**inspecteur** de l'objet *multislider* et notez les différents paramètres. Notre objet a un jeu de couleurs sélectionné et **4** niveaux de couleurs "candycane" alternées sélectionnées. Si vous verrouillez le patcheur et dessinez dans le multistrider, vous pouvez modifier la mélodie jouée dans la séquence. En cliquant sur l'objet *button* connecté à l'*uzi*, vous générerez une nouvelle mélodie de type *drunk*.

Multislider pour la visualisation des données

Examinons maintenant la zone du patch étiquetée **2**, qui est un patch de surveillance pour notre souris. Demarrez le *metro* en haut du patch avec le *toggle*, puis déplacez votre souris sur l'écran. Vous verrez que les deux zones de dessin en haut suivent les positions X et Y actuelles de la souris

(fournies par notre abstraction **WTHITM**), tandis que la zone de graphique plus grande en bas présente un affichage courant de quatre lignes, mis à l'échelle et modifié par la logique ci-dessus.

Dans les didacticiels précédents, nous avons utilisé l'objet *lcd* pour dessiner nos informations de suivi de manière créative. Cependant, dans ce cas, nous utilisons plutôt l'objet *multislider*. L'utilisation la plus courante de *multislider* est celle d'une banque de curseurs, mais il possède quelques modes qui sont utiles pour afficher un flux de données. Ces modes sont répertoriés en tant qu'options pour l'attribut **Slider Style** (dans l'inspecteur); ce sont les options **Scroll**, qui acceptent les valeurs entrantes, les ajoutent à la fin des données affichées, puis font défiler l'affichage pour les rendre visibles.

Les deux objets *multislider* plus petits situés en haut du patch affichent les données provenant de notre abstraction **WTHITM** (que nous avons utilisée dans des tutoriels précédents), ce qui permet de convertir les mouvements de la souris en nombres allant de **0,0** à **1,0**. Cette valeur est envoyée dans le *multislider* (pour être affichée), puis est sortie du *multislider* dans un patch qui produit quatre valeurs à partir de ces données (la position X, la position Y et une paire de coordonnées polaires pour la distance et l'angle de la souris par rapport au centre de l'écran). Le tout est envoyé dans le *multislider* inférieur, qui est à la fois graphiquement déroutant (rouge sur vert!) et particulier dans son affichage: il montre simultanément quatre flux de données.

Il serait très difficile d'essayer de faire quelque chose comme cela dans un objet *lcd*, mais pour réaliser un affichage multiligne dans un *multislider*, il suffit de dire qu'il y a quatre curseurs, ce qui permet de définir la plage d'affichage (dans ce cas, de **-1,0** à **1,0**).), et laisser l'objet faire fonctionner sa magie. Dans ce cas, les coordonnées X et Y de la souris sont utilisées pour créer quatre flux de données différents, chacun étant affiché dans sa propre section du graphique.

Surveillance des flux de données avec Watchpoints

Bien que le *multislider* soit utile pour afficher des flux de données numériques, Max possède également un certain nombre de systèmes intégrés permettant de surveiller les données. Si vous avez déjà un écran rempli d'objets et de cordons de raccordement, vous ne voudrez probablement pas utiliser la surface de l'écran pour un grand affichage graphique tel que *multislider*. Et si les données sont générées trop rapidement ou à partir de trop d'endroits, vous ne voudrez peut-être pas non plus *imprimer* des éléments sur la Max Console. Quelle est la meilleure option ? Les Watchpoints !

Un **watchpoint** est une pièce jointe à un cordon de raccordement qui vous permet de visualiser les messages qu'il transporte sans avoir à changer de cordon. Si vous faites un click droit avec la souris sur un cordon de raccordement et que vous faites apparaître le menu correspondant, vous trouverez deux options de watchpoint: «Add watchpoint - Monitor» et «Add watchpoint - Break». La première option définit un point de surveillance qui vous permet de contrôler les valeurs des messages, tandis que la seconde force le patch à s'arrêter dès qu'un message apparaît sur le cordon de raccordement sélectionné - nous avons utilisé cette option dans un tutoriel précédent pour examiner l'ordre des messages. Regardons quelques valeurs s'exécuter sur les cordons de raccordement.

Vous pouvez gérer et visualiser les watchpoints en sélectionnant **Watchpoints** dans le menu **Debug**. Cela fait apparaître une fenêtre avec une ligne pour chaque point de contrôle (identifié par les cercles colorés et numérotés sur les cordons de raccordement). Si vous démarrez l'objet *metro* dans l'une des sections du patch à l'aide des objets *toggle*, vous commencerez à générer des messages - et vous verrez les résultats dans la fenêtre watchpoint. La fenêtre et les colonnes sont redimensionnables. Par conséquent, si vous avez besoin de voir plus de données que celles qui sont

visibles, vous pouvez agrandir la fenêtre ou élargir les colonnes. Vous aurez probablement besoin d'augmenter la taille de la colonne **Valeur** pour voir toutes les valeurs du watchpoint **3**. Vous pouvez l'élargir en sélectionnant le séparateur d'en-tête entre les colonnes Valeur et Décompte et en le déplaçant vers la droite.

Les colonnes de la fenêtre Watchpoints s'expliquent d'elles-mêmes:

- La première colonne affiche l'**identifiant** du watchpoint.
- La deuxième colonne affiche le **type** d'action affecté au watchpoint.
- La troisième colonne affiche la **classe** de l'objet qui envoie le message.
- La quatrième colonne indique le **nom** du subpatcher contenant le watchpoint (ou est vide s'il se trouve sur le patch principal).
- La cinquième colonne affiche la ou les **valeurs** associées au dernier message reçu.
- La sixième colonne affiche le **nombre** de messages reçus par ce watchpoint.
- La septième (et dernière) colonne affiche un **historique** des trois dernières valeurs vues à ce watchpoint.

Si vous démarrez la section **2** de patcheur et déplacez votre souris avec la fenêtre watchpoints visible, vous verrez les trois premiers watchpoints suivre le mouvement de votre souris. Le format de la fenêtre (et l'emplacement des watchpoints) vous aide à voir rapidement que les coordonnées X et Y, visualisées avec les watchpoints **1** et **2**, sont des nombres à virgule flottante correctement mis à l'échelle de **0,0** à **1,0**. Vous pouvez également voir l'effet de l'objet *cartopol* et des objets mathématiques «diviser par pi» (/ **3.1415**) sur les coordonnées X / Y en visualisant les troisième et quatrième valeurs du watchpoint **3** (pilote par l'objet *pack*). Les watchpoints peuvent fournir un moyen simple et facile de vérifier que les résultats de vos calculs correspondent aux plages attendues.

Comme nous l'avons vu dans un précédent tutoriel, le passage d'un watchpoint à «break» ou à «break / watch» aura un effet spécial lorsque le **débogage** est activé: tout message vu par le watchpoint entraînera une pause du programme. Une fois le patch en pause, la **fenêtre de débogage** s'affiche et vous pouvez parcourir le patch, un message à la fois, en regardant le patch fonctionner pendant que la **fenêtre de débogage** affiche la trace de la pile de votre patcheur.

Résumé

Tout en ne révélant qu'un seul nouvel objet (*multislider*), ce didacticiel a introduit un certain nombre de techniques importantes pour la création et le suivi des valeurs dans un flux de données. Nous avons vu comment *multislider* dispose d'options d'affichage de données intégrées qui peuvent grandement simplifier le suivi des données numériques en temps réel. À des fins de débogage, nous avons également vu comment les points de surveillance peuvent être utilisés pour suivre n'importe quel message sur n'importe quel cordon de raccordement sans modifier notre programmation.