

29-Mise à l'échelle des données

introduction

Ce tutoriel est axé sur les problèmes de mise à l'échelle et de lissage des données. Il introduit l'objet *iter* pour les itérations de liste et les objets *trough*, *peak* et *slide* pour la mise à l'échelle et le lissage des données. Nous examinerons également les options supplémentaires d'affichage des données pour l'objet *multislider*, en particulier la possibilité de définir une plage minimale et maximale et l'utilisation du contrôle du maintien du peak-hold.

Différents objets Max et périphériques physiques génèrent des valeurs de différentes plages et de différents formats. L'une des tâches que vous allez souvent effectuer consiste à modifier le format et l'échelle d'entrée dans un ensemble de données pour qu'ils correspondent au format et à l'échelle nécessaires pour votre sortie. Bien que les didacticiels précédents aient décrit l'utilisation de l'objet *scale* pour cette fonction, il peut être nécessaire d'utiliser d'autres objets pour d'abord trouver la plage numérique pour notre mise à l'échelle. Les objets présentés dans ce didacticiel offrent de nouvelles options pour la mise à l'échelle, de lissage et d'affichage des données afin de répondre à vos besoins de programmation.

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

Examen de nouveaux objets

Jetez un coup d'œil à notre patcheur de tutoriel: sur le côté gauche du patcheur se trouve un aperçu de quelques nouveaux objets (soulignés en vert). Le patch le plus en haut montre l'objet *iter* en cours d'utilisation - il accepte une liste et sort les valeurs individuelles dans l'ordre. Lorsque nous cliquons sur la boîte de **message**, la liste de quatre valeurs est envoyée à *iter*, qui génère quatre messages, chacun contenant un élément de liste. C'est le moyen le plus simple de transformer une liste en un flux de données et (comme nous le verrons bientôt) c'est utile pour atomiser le contenu d'une liste afin de déclencher un nouvel événement à partir de chaque élément de la liste.

Le subpatch suivant contient des exemples des objets *trough* et *peak*. Ils servent des fonctions complémentaires: les deux objets examinent un flux de nombres entrants, *trough* capturant (et sortant) la plus petite valeur reçue jusqu'à présent, tandis que *peak* capture et envoie la valeur la plus élevée. Lorsque nous modifions la valeur dans la boîte de *nombre*, nous voyons que les objets capturent les valeurs basses et hautes, avec une nouvelle sortie se produisant chaque fois qu'une nouvelle valeur haute ou basse est atteinte.

Enfin, le subpatch du bas montre trois utilisations de l'objet *slide*. L'objet *slide* est utilisé pour lisser les données en fonction d'un «facteur de lissage», avec des facteurs distincts pour les mouvements **à la hausse** et **à la baisse**. La fonction du facteur de lissage est plutôt simple: il sert à diviser la différence entre le dernier message reçu et le message actuel, ce qui réduit la quantité de «swing» dans toute transition de valeur. Les facteurs de lissage sont fournis sous la forme de deux arguments à l'objet: le premier argument est le facteur de mouvement ascendant et le second argument est le facteur de mouvement descendant.

Par exemple, lorsque vous démarrez le patch pour la première fois, la «dernière» valeur est **0.0**. Si vous effectuez une transition immédiate vers **0.5** (ce que vous pouvez faire en entrant une valeur dans la boîte de *nombre* reliée aux objets *slide*), vous verrez que le premier curseur se déplace sur **0,01**. C'est parce qu'il y eu une transition vers le haut de la valeur ; donc premier facteur (**50**) a été utilisé: la différence entre la valeur initiale de **0,00** et **0,50** est de **0,50**; cette différence est divisée

par le facteur (**50.**) pour donner **0,01**.

Les trois objets *slide* présentent différents facteurs de lissage: le premier fournit des facteurs égaux pour les mouvements ascendants et descendants, le second n'a pas de lissage ascendant (un facteur de **1,0**) et le troisième a un faible facteur descendant (un facteur de **10,0**). Lorsque vous déplacez le curseur d'entrée, vous verrez que chaque *multislider* de sortie répond différemment, lissant la sortie en fonction des facteurs de lissage ascendants et descendants de chaque objet *slide*.

La vue d'ensemble du patch principal

Le patch principal utilise les données météorologiques comme matériau source; elles pilotent l'affichage dans notre *multislider* et sont également utilisées pour créer des accords de quatre notes à lire sur le synthétiseur MIDI interne de l'ordinateur. Double-cliquez sur les deux objets *coll* dans le patch pour vous familiariser avec leur contenu - tous deux contiennent douze lignes de données, la sortie de l'un servant d'index à l'autre. Lorsque vous démarrez le *metro* en haut du patch avec l'objet *toggle*, les messages **bang** sont utilisés pour générer des informations de calendrier- plus précisément, la sortie d'un compteur (qui compte les jours) est utilisée pour interroger une petite base de données *coll* qui produit les mois en fonction des jours de l'année civile où ils commencent - par exemple le **90ème** jour de l'année est le premier jour du mois **4** (avril). Cette valeur mensuelle (comprise entre **1** et **12**) est ensuite utilisée pour interroger la base de données *coll* chargée à partir du fichier *weather.txt*, qui contient les températures mensuelles moyennes de quatre villes (New York, Paris, Buenos Aires et Tokyo). Ces données sont décompressées (puisqu'elles sortent du **coll** sous forme d'une liste) et sont envoyées à quatre objets *float* (abrégiés "**f**"). Le *metro* produit ensuite un **bang** sur les objets *float* pour créer un flux de sortie. Il s'agit d'une bonne technique à utiliser dans les systèmes où vous souhaitez que quelque chose se produise sur une horloge régulière mais où les données entrantes ne sont pas produites à une vitesse constante - dans notre exemple, la liste des températures ne sort de l'objet *coll* que lorsque le mois change.

La sortie des objets *float* est envoyée aux objets *slide* avec un facteur de lissage de **15.0** pour les mouvements à la hausse et à la baisse. Cela signifie que, même si la même valeur peut être envoyée jusqu'à 31 fois (puisque'il y a jusqu'à 31 jours dans un mois), la valeur changera quand la sortie lissée atteindra la valeur réelle. Les sorties des quatre objets *slide* sont regroupées et envoyées dans le *multislider*, où elles sont affichées. Cet ensemble de données nouvellement créé est également sorti du *multislider*, où il est divisé (par l'objet *iter*) et utilisé pour créer quatre notes pour la lecture du synthétiseur.

Ce *multislider* utilise quelques nouvelles techniques d'affichage des données. Ouvrez l'inspecteur d'objets pour le *multislider* et vous verrez que l'attribut **Style de curseur** est défini sur **Bar**, ce qui nous donne la barre pleine. L'option **Candycane** est également activée pour que les barres soient colorées en fonction des couleurs de l'inspecteur. Enfin, une option de **Peak hold** est définie, ce qui entraîne l'apparition d'une petite barre au sommet du point le plus élevé atteint par la valeur. Cela nous donne un indicateur visuel de la valeur maximale observée pour chacune des quatre villes.

Un *toggle* relié à une boîte de *message* permet de désactiver ou d'activer la fonction de maintien de la valeur maximale; c'est un bon exemple d'utilisation de messages de type inspecteur pour contrôler la fonction d'un objet. Comme toute modification de l'attribut **peakhold** doit réinitialiser le suivi des peaks, un message complémentaire (**peakreset**) réinitialise cette valeur chaque fois que le *toggle* est modifié.

La section min / max

- Ouvrez le **setup** de l'objet *patcher* en double-cliquant dessus.

L'autre partie de ce patch qui mérite d'être examinée est la configuration effectuée juste à gauche du patch principal. Cette section de patch a un but unique: configurer le *multislider* pour une mise à l'échelle correcte de l'affichage. Un message **bang** (soit envoyé en cliquant sur le *button*, soit généré par un *loadbang*) commence par définir les objets *trough* et *peak* avec des seuils très élevés et très bas, respectivement - cela garantit que toute nouvelle valeur qui leur est envoyée définira automatiquement une nouvelle norme dans l'objet. Le **bang** déclenche ensuite un message **dump** vers l'objet *coll*, ce qui entraîne l'envoi de toutes les entrées de l'objet, une liste à la fois.

Au fur et à mesure que les listes sont envoyées, elles sont traitées par un objet *iter*, qui les divise en messages à valeur unique. Ceux-ci sont examinées par les objets *trough* et *peak*, qui trouvent les valeurs les plus basses et les plus hautes dans l'ensemble de données (**32** et **81**, respectivement). Lorsque l'une ou l'autre de ces valeurs est trouvée, l'objet *pak* génère une liste de valeurs, que *prepend* joint au message *setminmax* envoyé au *multislider*. Cela modifie la plage d'affichage qui sera présentée, ce qui amène le *multislider* à limiter son affichage à la plage réelle de températures hautes et basses qui sera émise par le *coll*. De cette façon, l'affichage du *multislider* est automatiquement calibré aux températures stockées dans le fichier *weather.txt*.

Résumé

Dans ce didacticiel, nous avons vu comment analyser des flux de données (en utilisant *iter*, *trough* et *peak*) et comment lisser leur sortie (en utilisant des *sliders*). Nous avons également vu quelques modèles de programmation qui peuvent être utiles pour configurer automatiquement des éléments d'interface utilisateur et pour sortir des données lorsque de nouvelles données sont envoyées moins fréquemment que nécessaire. Enfin, nous avons exploré plus en détail l'objet flexible *multislider*, en l'utilisant pour un affichage à barres avec des valeurs de maintien de crête et un affichage en couleur.