

3: Nombres et Listes

Données numériques dans Max

Une grande partie de la programmation dans Max implique la manipulation de différents types de données. Dans ce tutoriel, nous verrons comment Max gère certains nouveaux types de messages: les nombres et les listes. Nous allons passer du temps avec les éléments de l'interface utilisateur qui nous permettent de manipuler des nombres et d'apprendre à utiliser des arguments pour insérer des éléments de liste dans un message.

Alors que les messages composés de texte sont utiles pour communiquer avec les humains (et envoyer des commandes à de nombreux objets Max que nous verrons plus loin), une grande partie du monde des communications dépend de nombres, qu'il s'agisse de données de notes MIDI ou de pixels vidéo, couleurs ou de contrôle de mouvement robotique. Maîtriser les messages numériques est essentiel pour comprendre le fonctionnement de Max.

Les messages numériques dans Max ne sont pas toujours des entités singulières; elles sont souvent combinées avec du texte de commande ou d'autres informations pour créer une liste de données utile à d'autres objets, à l'utilisateur de votre patch ou au monde extérieur. Il existe donc plusieurs méthodes pour combiner des messages en listes.

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

Boîtes de nombres

Regardez le patch du tutoriel. La fenêtre du patch contient plusieurs petits patches qui permettent de manipuler des nombres et des listes. Le patch en haut à gauche porte sur un nouvel objet appelé **boîte de nombre**. Il traite des nombres *entiers* (souvent appelés nombres entiers) et accepte de nombreux messages différents. Cliquez sur certaines des boîtes de *message* connectées et notez leur effet sur la boîte de nombres ainsi que sur les messages qui apparaissent dans la Max Console via l'objet d'*impression*.

L'effet le plus notable se produit lorsque des données numériques sont envoyées à la boîte de nombres: elle réagit en affichant les données et en envoyant le nombre depuis sa sortie. Lorsqu'un *nombre à virgule flottante* tel que 3.5 est envoyé à l'objet, il prend la partie entière du nombre et ignore la partie décimale. Si un message non numérique tel que "Hey!" est envoyé, la Max Console affiche une réclamation de l'objet, indiquant qu'il ne comprend pas ce type de message.

Le patch à droite est similaire, mais utilise une boîte de *nombre à virgule flottante* qui permet aux valeurs décimales d'être affichées et sorties. La plupart des messages produisent des résultats de la même manière que la boîte de nombres entiers. Toutefois, lorsqu'une valeur en virgule flottante est reçue, la valeur décimale est enregistrée et affichée dans cette boîte de nombres. Si la case n'est pas assez large pour afficher le nombre entier, seule une partie de celle-ci peut être affichée, mais tout est là (et transmis). Vous pouvez ajuster le nombre de décimales affichées dans l'inspecteur.

Deux messages spéciaux sont acceptés par la *boîte de nombre* ou la *boîte à virgule flottante*. Tout d'abord, l'envoi d'un **bang** dans une boîte de nombres force la sortie de la valeur actuelle sans aucune modification des données. Deuxièmement, l'inclusion d'un nombre dans un message "set" modifiera les données sans aucune sortie, ce qui en fait un moyen de changer en silence le contenu d'une boîte de nombres. La méthode set est assez courante dans les objets Max comme moyen pour manipuler l'état d'un objet sans déclencher de messages de sortie.

*L'interface utilisateur de la **boîte de nombres***

Outre l'envoi de messages aux boîtes de *nombres* ou aux boîtes de message à *virgule flottante*, nous pouvons également saisir et manipuler directement les valeurs stockées. Avec le patcheur verrouillé, sélectionnez la boîte, tapez une valeur et appuyez sur retour ou cliquez ailleurs. De plus, nous pouvons cliquer-glisser dans la boîte pour modifier manuellement le contenu. Ceci est utile pour les situations de performance où nous pouvons vouloir changer un nombre en temps réel pendant l'exécution du patch. Plutôt que de devoir configurer des boîtes de message pour chaque valeur possible dont nous pourrions avoir besoin, nous pouvons manipuler la boîte de nombres pour modifier un paramètre. Si nous cliquons-glissons sur la boîte de nombres entiers, nous voyons que la Max Console affiche un flux de données - la boîte de nombres génère une sortie pour la plupart des valeurs que nous faisons défiler. Elle en saute cependant quelques-unes, selon la vitesse à laquelle vous faites glisser le curseur. Pour inclure chaque valeur, sélectionnez la boîte et appuyez sur la flèche vers le haut ou vers le bas.

Avec les données entières, il est facile de prendre en charge une zone de valeur de défilement - le déplacement vers le haut ajoute 1 à la valeur, tandis que le défilement vers le bas soustrait 1. Cependant, que devons-nous ajouter / soustraire lorsque nous avons des valeurs à virgule flottante ? Dans le cas de la boîte de valeurs à virgule flottante, la réponse dépend de l'endroit où nous plaçons la souris avant de cliquer-glisser. Si la souris est sur la partie entière des données (à gauche de la virgule), cliquez-glissez pour produire des changements identiques à la boîte de *nombres* entiers. En revanche, si nous positionnons la souris sur la partie fractionnaire des données, nous pouvons modifier n'importe quel des chiffres affichés. (Tous les chiffres situés à droite de la souris sont réglés sur 0.) Cela nous permet de manipuler les données stockées dans la boîte de nombres à virgule flottante avec un grain "grossier" ou "fin".

Le troisième patch contient des boîtes de *nombres* et de nombres à *virgule flottante* reliées les unes aux autres. Ici, vous pouvez voir la différence visible entre les boîtes de nombres entiers et les nombres à virgule flottante - les versions à virgule flottante ont un point décimal affiché. Si vous modifiez la boîte de nombres la plus élevée (nombres à virgule flottante), vous verrez que ses données sont utilisées pour mettre à jour une boîte de nombres entiers, laquelle met ensuite à jour la boîte de nombres (à virgule flottante) suivante. Vous pouvez également manipuler directement la boîte de nombres entiers pour voir comment ses messages affectent la variante à virgule flottante.

Saisissez un grand nombre dans la boîte de nombre située en haut, quelque chose comme 60000. Vous verrez que les boîtes de nombres entiers ne peuvent pas afficher ce nombre, car il est trop grand pour la zone d'affichage prévue. Déverrouillez le patch et déplacez votre souris *vers la droite* de l'objet. Vous verrez apparaître une petite "poignée" et le curseur se changera en flèche gauche. Cela s'appelle une *boîte de croissance* et peut être utilisé pour redimensionner tous les objets dans Max. Cliquez-glissez sur la poignée pour changer la largeur de la boîte de nombres, puis cliquez sur une zone vide du patch pour désélectionner l'objet. Vous verrez un changement immédiat dans l'affichage des nombres entiers.

La taille des nombres entiers pouvant être manipulés dépend du fait que Max fonctionne en mode 32 ou 64 bits. Pour voir la différence, entrez un nombre ridiculement élevé, tel que 7245569558. La boîte des nombres entiers peut avoir changé en -2147483648. Comment est-ce arrivé ? Les nombres entiers ont toujours une limite - en mode 32 bits, la valeur maximale d'un entier est 2147483520. En mode 64 bits, la valeur supérieure est 9223372036854775807. Tous les nombres supérieurs à cette valeur entraînent un "débordement" et deviennent le plus grand nombre négatif. D'autres valeurs très élevées peuvent fournir d'autres valeurs, en fonction du dépassement de cette limite par le nombre. La plage des nombres à virgule flottante est d'environ 38 chiffres en mode 32 bits et de 308

chiffres en mode 64 bits.

Examiner les messages avec des arguments remplaçables

Le patch central est un peu différent - il ne s'agit que d'un groupe de boîtes de *messages* dont le contenu est constitué de groupes de nombres et de mots (appelés **symboles** dans Max). Lorsque nous cliquons sur la boîte de message avec "60 30", nous voyons les deux nombres affichés ensemble sur la Max Console. Un message contenant plusieurs nombres ou symboles est appelé une **liste**; la liste est un mécanisme permettant de regrouper les données en un seul message pouvant être envoyé via des cordons de raccordement à d'autres objets. Les nombres peuvent être des nombres entiers, des nombres à virgule flottante ou une combinaison, comme vous pouvez le voir dans le deuxième exemple "22 33.9 -5 -44.2".

L'exemple de message suivant est une boîte de *nombres* connectée à une boîte de *messages*. Utilisez la souris pour modifier la valeur de la boîte de nombres. Le résultat est que la console Max affiche le message "puces", mais remplace le \$1 par le nombre entrant. Le texte "\$1" est appelé un **argument remplaçable** ou **jeton** et vous permet de substituer le contenu du message de manière variable en utilisant une boîte de message.

Le dernier exemple de ce patch accepte deux valeurs (comme l'indiquent les jetons \$1 et \$2) et utilise les valeurs de la liste entrante pour créer le message "puces et tiques". Vous pouvez avoir jusqu'à 9 arguments remplaçables, numérotés de **\$1** à **\$9**, et les arguments peuvent être dans n'importe quel ordre. Essayons une expérience avec les arguments:

Déverrouillez la fenêtre du patch et sélectionnez le texte dans la dernière boîte de *message*. À la fin du texte existant, ajoutez le texte suivant après un espace: "les ticks à \$2 sont pires que les puces à \$1". Verrouillez le patch et cliquez sur l'une des deux listes connectées. La nouvelle instruction devrait s'afficher dans la Max Console, le contenu de la liste approprié remplaçant les arguments du message. Cela démontre que les arguments remplaçables peuvent être réutilisés (vous pouvez avoir plusieurs arguments \$1 dans un message) et peuvent être appelés dans n'importe quel ordre (par exemple, \$2 peut précéder \$1 dans un message).

Construire des listes avec pack et pak

Les patches dans la section inférieure montrent une méthode alternative pour construire des listes. L'objet *pack* aura une entrée pour chaque élément dans ses arguments (qui peuvent être des symboles ou des nombres). Les arguments constituent le message par défaut qui sera envoyé si l'objet est frappé par un bang. Les données reçues dans n'importe quelle entrée remplacent l'élément associé dans la sortie. Toutefois, le type des données entrantes est modifié pour correspondre à l'argument original. Dans le patch "lists made with pack", la troisième entrée est associée à un argument entier, mais une boîte flottante est attachée. Ces nombres seront convertis. (Lorsque des arguments de symbole et des entrées numériques, ou inversement, sont mélangés, les résultats sont des zéros ou des espaces vides.) Les données reçues dans l'entrée de gauche remplacent le premier argument et envoient la liste. Si une liste de données est envoyée à une entrée quelconque d'un objet *pack*, le premier élément remplace l'argument associé à l'entrée et les éléments suivants sont appliqués aux entrées suivantes.

L'objet *pak* (prononcé "pock") est exactement comme un *pack* avec une différence. La sortie est envoyée lorsque les données sont reçues à n'importe quelle entrée. Modifiez certaines boîtes de nombres dans le patch "Listes créées avec pak" avec la souris et observez le nombre de messages affichés dans la fenêtre Max. *Pack* est utilisé lorsque des nombres doivent être rassemblés dans une liste et envoyés lorsqu'ils sont prêts, *pak* est utilisé lorsque des listes doivent être mises à jour dès

qu'une valeur est reçue.

Résumé

Max peut envoyer des données numériques sous forme de messages, tout comme du texte. Quelques objets d'interface utilisateur nous aident à construire et à traiter des données numériques, tels que les boîtes de *nombres* entiers et les nombres flottants. Max peut également conserver des groupes de nombres liés dans un seul message à l'aide de listes. Les boîtes de *messages* peuvent être utilisées pour construire des listes et d'autres messages plus complexes avec des arguments remplaçables. Les listes peuvent également être construites avec *pack* et *pak*.