

38-Stockage de patcheur

introduction

Max dispose d'une grande variété d'objets pour stocker et gérer des données. Les objets tels que *table*, *funbuff*, *coll* et *umenu* stockent toutes les informations que vous pouvez définir manuellement ou via un mécanisme au sein même d'un patch. L'objet *preset*, qui stocke également des données, est utile pour capturer et rappeler les états des objets d'interface utilisateur dans un patcheur. Ce qui vous permet de restaurer rapidement les paramètres de plusieurs parties de votre patch en une seule étape.

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

Qu'est-ce que *pattr* ?

Souvent, un projet dans Max nécessite un système de gestion d'état plus sophistiqué que ne le permettent les objets de structures de données de base. La famille d'objets *pattr* (*pattr*, *pattrhub*, *autopattr* et *pattrstorage*) fournit la fonctionnalité de l'objet *preset* avec un ensemble plus étendu de caractéristiques de gestion d'état :

On peut obtenir des ensembles de données d'objets à travers une hiérarchie de patches Max. Par exemple, avec *pattr*, vous pouvez contrôler l'état des objets à l'intérieur des objets patcher et bpatcher depuis le niveau supérieur du patch.

Nous pouvons stocker tout type de données utilisées par tout type d'objet (y compris des données simplement stockées à l'intérieur d'un *pattr* lui-même, sans aucun autre objet impliqué).

Nous pouvons utiliser des objets *pattr* pour définir et interroger à distance l'état des objets contrôlés par le système *pattr* depuis n'importe quel endroit du patch.

Les objets *pattr* stockent des groupes de paramètres sous forme de fichiers XML, ce qui nous permet de lire et de modifier facilement des données enregistrées en dehors de Max.

Les objets *pattr* peuvent rappeler l'état des objets dans un ordre spécifique, en évitant les difficultés liées, par exemple, au rappel d'un objet *toggle* qui démarre un processus avant que toutes les variables ne soient en place pour que le processus fonctionne correctement.

Non seulement nous pouvons stocker les états de nombreux objets sous une seule adresse, mais nous pouvons également interpoler entre ces états, ce qui permet un fondu enchaîné transparent entre plusieurs paramètres.

Les objets *pattr* présentent une interface de haut niveau permettant de visualiser et de gérer l'état actuel des objets contrôlés et les états qui ont été enregistrés.

Dans ce didacticiel, nous examinerons certaines des fonctions de base de *pattr* et *pattrhub*, en montrant comment les objets peuvent être utilisés pour communiquer à distance avec des objets situés n'importe où dans un patch Max.

objet *Binding*

Jetez un coup d'œil au patcheur du tutoriel. Activez l'objet *metro* avec la boîte d'objet *toggle* étiquetée "start". Vous devriez entendre de la musique si vous double-cliquez sur l'objet *noteout* et sélectionnez un synthétiseur de sortie valide. La boîte de *nombre* attachée à l'objet *metro* contrôle la vitesse du motif généré. Cliquez sur les boîtes de *messages* colorées à droite du patch et notez qu'elles modifient la valeur de cette boîte de *nombre* même si elles ne sont pas directement connectées à celle-ci.

Le fait de cliquer dans une boîte de *message* colorée modifie la valeur temporelle de l'objet *metro*.

L'objet *pattr* situé à droite de l'objet *metro* contrôle la valeur dans la boîte de *nombre* située au-dessus de celui-ci. Dans la langue de stockage des patcheurs, nous disons que notre objet *pattr* est lié à la boîte de *nombres*. Lorsque nous envoyons des messages à cet objet *pattr*, il se réfère à tout moment aux données stockées dans la boîte de *nombres*, ce qui nous permet de demander à l'objet *pattr* quel est la valeur actuelle du nombre, ou de lui demander de régler le nombre sur une valeur spécifique à distance.

Notre objet *pattr* a un nom - **speed** - qui est défini par l'argument de l'objet et permet à d'autres objets de la famille *pattr* de communiquer avec lui. L'objet *pattrhub* situé au bas du patch peut recevoir des messages et les transférer à tous les objets *pattr* du patch. En envoyant le message **speed 125** au *pattrhub*, nous demandons à l'objet de trouver le *pattr* portant le nom **speed**. S'il trouve l'objet *pattr* correspondant, il lui demandera d'envoyer le message **speed 125** à l'objet auquel il est lié (dans ce cas, notre boîte de *nombres*).

Déverrouillez le patch et cliquez sur la boîte de *nombres* liée à l'objet *metro*. Sélectionnez **Nom...** dans le menu **Objet**. L'inspecteur d'objets s'ouvre et montre que le **nom de script** de la boîte de *nombres* est **temponum**.

Notre objet *pattr* est lié à notre boîte de *nombres* car l'attribut **bindto** de notre objet *pattr* est défini sur le nom de notre boîte de *nombres* (**temponum**). Le fait de **nommer** les objets Max nous permet d'y accéder plus facilement via le système *pattr*. Notez que l'objet *pattrhub* contrôle notre boîte de *nombres* via le nom de l'objet *pattr* auquel il est lié (**speed**), et non le nom de la boîte de *nombres* elle-même (**temponum**).

Renommez la boîte de *nombres* en quelque chose d'autre et fermez l'inspecteur. Essayez de modifier les paramètres du patch en cliquant à nouveau sur les boîtes de messages.

Notez que si nous renommons notre objet référé, notre objet *pattr* ne peut plus se lier à lui. Pour le lier à nouveau, nous devons dire à *pattr* de se lier à nouveau à notre boîte de *nombres*.

Ouvrez l'inspecteur pour la boîte de *nombres* et renommez-le **temponum**. Fermez l'inspecteur et recréez l'objet *pattr* lié au **temponum** (soit en supprimant l'objet et en annulant la suppression, soit en créant un nouvel objet *pattr*). Vérifiez qu'il reçoit les mises à jour de l'objet *pattrhub* en cliquant à nouveau sur les boîtes de *messages*.

Au lieu de recréer l'objet *pattr*, nous aurions pu envoyer à notre nouvel objet *pattr* «non lié» un message **bindto** avec le nom de la boîte de *messages*, en définissant l'attribut **bindto** sur un objet valide.

Plus d'une façon

Nommer un objet Max et y lier explicitement un *pattr* avec son attribut **bindto** n'est qu'une façon de faire en sorte qu'un *pattr* fasse référence à un objet dans notre patch. Elle présente l'avantage de ne pas nécessiter de cordons de connexion entre les deux objets, ce qui leur permet de communiquer à partir de parties distantes du patcheur.

Ouvrez le subpatch **notes** en double-cliquant sur l'objet *patcher* qui le contient. Ce subpatch contient huit objets boîte de *nombres* liés à l'objet *pattr* qui définissent les quatre hauteurs dans notre générateur de motifs, ainsi que le nombre de battements entre chaque déclenchement de la hauteur. C'est la partie du patch qui nous permet de construire un motif polyrythmique à partir d'un seul objet *counter* (piloté par l'objet *metro*) dans le patcheur principal.

Cliquez sur les boîtes de *messages* dans le patcheur principal avec cette fenêtre ouverte et notez que les objets de boîte de *nombres* du subpatch se mettent à jour correctement lorsque vous cliquez sur les différents motifs. Le fait de cliquer sur la boîte de *message* met également à jour les valeurs du subpatch !

En plus de lier des objets à *pattr* par le biais d'une appellation explicite et de l'attribut **bindto**, les objets *pattr* peuvent contrôler d'autres objets Max par le biais des connexions patchcord.

Le subpatch *notes* montre deux manières possibles de le faire:

1. Utiliser l'objet *pattr* lui-même pour stocker les données (*vals*)
2. **Liaison** automatique d'un objet *pattr* à un objet à l'aide de la sortie centrale de cet objet (*mods*).

Stockage interne

Si un objet *pattr* est non lié, il stockera tous les messages envoyés dans son entrée gauche et les rappellera depuis sa sortie gauche. Les messages envoyés à un objet *pattr* non lié à partir d'un objet *pattrhub* seront également renvoyés par la sortie de gauche.

Déverrouillez le subpatch *Notes* et attachez un objet *slider* à la sortie gauche de l'un des objets *pattr* nommés **val_a**, **val_b**, **val_c** ou **val_d**. Rappelez l'état des objets *pattr* en cliquant sur les boîtes de *messages* dans le patch principal. Faites défiler la boîte de *nombre* attachée à l'objet *pattr*. L'objet *slider* devrait également bouger.

Les objets *pattr* contrôlant nos hauteurs ne sont pas liés à un seul objet et conservent les valeurs des objets de la boîte de *nombres* qui leur sont attachés, non pas en faisant directement référence aux objets, mais en recevant leurs valeurs sous forme de messages. En conséquence, les objets boîte de *nombres* attachés à ces objets *pattr* peuvent être supprimés et remplacés par d'autres objets (ou plus d'un objets). Comme l'objet *pattr* émet ses valeurs lorsqu'il est modifié (directement ou à distance), nous pouvons facilement attacher plusieurs objets à l'objet *pattr* pour définir et afficher les valeurs stockées dans l'objet.

L'utilisation de l'objet *prepend* avec l'argument 'set' dans les connexions *pattr* permet d'empêcher le déclenchement d'événements MIDI lorsque nous rappelons l'état de l'objet *pattr*. Contrairement à la plupart des objets Max, les objets *pattr* peuvent être connectés à d'autres objets qui leur sont directement connectés à leur tour sans provoquer de débordement de pile.

Liaison automatique

Dans le même subpatch, essayez de connecter un objet *slider* à la sortie centrale de l'un des objets *pattr* étiquetés **mod_a**, **mod_b**, **mod_c** ou **mod_d**. La console Max devrait afficher un message d'erreur et le programme devrait vous empêcher d'établir la connexion.

La sortie médiane de *pattr* permet de lier automatiquement le *pattr* à un seul objet dans le même patcher. Les objets boîte de *nombres* contrôlant les objets % communiquent avec leurs objets *pattr* respectifs, exactement comme si nous les avions nommés et liés manuellement aux objets *pattr*.

Examinez les noms des objets de boîte de *nombres* en sélectionnant **Nom...** dans le menu **Objet**.

La sortie **bindto** (milieu) des objets *pattr* donne automatiquement un nom à l'objet de référence s'il n'en a pas déjà un. Le nom est basé sur la classe d'objet attachée, et plusieurs objets de la même classe dans un patch recevront une notation incrémentielle pour les identifier de manière unique (par exemple, **nombre**, **nombre [1]**, **nombre [2]**, etc.).

De retour dans le patch principal, ouvrez le subpatch **durs** du patch principal en double-cliquant sur l'objet patcher qui le contient. Observez que les deux objets *pattr* qu'il contient (min et max) ont été liés à deux objets boîtes de *nombres* par leurs sorties centrales.

Un hub pour les gouverner tous

Fermez les subpatches et retournez au patch principal du didacticiel. Examinez à nouveau les boîtes de message et remarquez la notation utilisée pour faire référence aux objets *pattr* dans les subpatches.

Comme la plupart de nos objets *pattr* sont à l'intérieur de subpatches, l'objet *pattrhub* s'y réfère en utilisant une notation séparée par des points, qui prend la forme suivante :

subpatcher: pattr_name

Nous pouvons communiquer avec les objets *pattr* dans des subpatches imbriqués selon la même convention, par exemple. un *pattr* nommé **yikes** dans un patcher appelé "**stuff**" dans un *bpatcher* appelé "**things**" serait accessible via *pattrhub* avec le nom suivant :

things: stuff: yikes

À l'aide de l'objet *umenu* du patch principal du didacticiel, sélectionnez l'un des objets *pattr* dans le patch.

En plus de définir l'état des objets *pattr* via *pattrhub*, nous pouvons également interroger l'état des objets *pattr*, qui indiqueront alors à *pattrhub* leur état interne ou la valeur de l'objet auquel ils sont liés. Pour ce faire, nous envoyons à *pattrhub* le message **get** suivi du nom de l'objet *pattr* que nous voulons interroger, sans espace dans le message, par exemple **getspeed** interrogera l'état du *pattr* nommé **speed**.

La boîte de *nombres* situé à droite de l'*umenu* affiche la valeur actuelle du *pattr* sélectionné, sortie par la deuxième sortie de *pattrhub* sous forme d'un message commençant par le nom du *pattr* interrogé.

Changez la valeur du *pattr* sélectionné en changeant la boîte de *nombre* à côté du *umenu*. Vérifiez que la valeur a changé en regardant la boîte de *nombre* contrôlée par ce *pattr*. Sélectionnez d'autres objets *pattr* et modifiez leurs valeurs, afin de vous familiariser avec la façon dont le système vous permet d'interroger et de définir l'état des objets n'importe où dans le patch.

Résumé

Les objets *pattr* fournissent un moyen puissant de gérer les données de plusieurs objets dans un patch Max. L'objet *pattr* peut conserver ses propres informations internes (envoyées et renvoyées par son entrée et sa sortie les plus à gauche), ou il peut se lier à un objet de l'interface utilisateur Max soit en nommant cet objet et en utilisant l'attribut *bindto* de l'objet *pattr*, soit en utilisant la sortie centrale de l'objet *pattr* pour nommer et se lier automatiquement à un objet. Chaque objet *pattr* ne peut être lié qu'à un seul objet à la fois. L'objet *pattrhub* vous permet de contrôler l'état de tous les objets *pattr* dans un patch, y compris ceux des subpatches, auxquels on peut accéder par un système de notation hiérarchique de noms de patcher successifs séparés par deux points (:) suivis du nom de l'objet *pattr* approprié. En envoyant un message **get** (suivi du nom d'un *pattr* sans espace intermédiaire), vous pouvez interroger l'état de n'importe quel objet *pattr* dans un patch à partir d'un *pattrhub*.