

9: Dessin à la souris

introduction

Dans les tutoriels précédents, nous avons examiné un ensemble d'objets de base que nous utilisons tout le temps pour construire des patchers dans Max. Dans ce tutoriel, nous allons utiliser ces objets (et quelques autres) pour contrôler réellement quelque chose (dans ce cas, le dessin). Nous allons examiner quelques façons de manipuler les messages en les acheminant (à l'aide de l'objet *gate*) et en manipulant des listes (à l'aide des objets *pack* et *unpack*). Nous interrogerons également l'état de la souris pour modifier le fonctionnement de notre patch. Nous en apprendrons davantage sur l'obtention d'informations sur des objets individuels et la modification de l'apparence des objets et des cordons de raccordement. Nous allons utiliser ces nouveaux objets et techniques pour contrôler l'objet *lcd*, ce qui nous permet de dessiner des graphiques simples en deux dimensions dans Max.

L'un des aspects les plus puissants de Max est la possibilité de créer de petits segments de programmation exécutant des fonctions spécifiques et de pouvoir acheminer des messages vers la partie appropriée d'un patcheur plus important. Le routage intra-patch est la clé de cette puissance, et l'objet *gate* fournit cette fonctionnalité de manière très flexible. Il est également utile de créer des messages complexes à la volée afin de pouvoir contrôler les processus de patch complexes avec une relative facilité.

Les objets *pack* et *unpack* sont utilisés pour créer des messages de type liste et pour les scinder à nouveau en leurs composants individuels. Les listes sont la clé pour travailler avec des objets qui nécessitent des messages plus complexes. Il est souvent nécessaire de construire et de déconstruire des listes à partir de composants plus petits (nombres entiers et nombres à virgule flottante, par exemple).

Enfin, nous chercherons à obtenir plus d'informations sur les messages des objets. Nous allons également apprendre à colorer et à manipuler les cordons de raccordement. Les indices visuels nous aident à comprendre la fonctionnalité des patchs des autres utilisateurs et à rendre les patchs plus lisibles lorsque nous partageons nos patchs.

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

Obtenir plus d'informations

Avant de commencer à regarder ce que fait notre patcheur de didacticiel, faisons une petite diversion pour voir comment nous pouvons obtenir des informations sur les objets que nous avons dans notre patch.. Déverrouillez le patcheur et déplacez votre souris autour des objets du patch. Vous avez probablement remarqué que lorsque nous plaçons notre souris sur une entrée ou une sortie, une petite boîte s'affiche contenant des informations sur l'utilisation de cette entrée ou de cette sortie. Ce texte d'aide sert à rappeler au programmeur ce que fait cette entrée ou cette sortie dans un objet. Vous remarquerez également qu'un petit cercle s'affiche autour de l'entrée/sortie.

Si nous passons la souris sur le bord gauche d'un objet, nous voyons un petit cercle avec un triangle. En cliquant dessus, vous ouvrez le **menu Action de l'objet**, qui affiche (via des sous-menus) l'état des attributs de l'objet, tous les messages entrants pris en charge, ainsi que des raccourcis pour afficher le fichier d'aide de l'objet, les pages de référence et même rechercher dans notre système des correspondances pertinentes dans le navigateur de fichiers. Cela permet d'obtenir un niveau de détail supérieur si nous avons besoin de connaître les détails de l'utilisation exacte de l'objet. Ces informations sont également disponibles dans les segments Messages et Attributs du menu

circulaire.

Il existe d'autres options pour les informations à la volée. Sélectionnez **Clue Window** dans le menu **Window** de Max; une petite fenêtre s'affiche. Maintenant, lorsque vous passez la souris sur un objet, vous verrez une brève explication de la fonction de l'objet s'afficher. L'utilisation de la fenêtre d'indices pour afficher l'utilisation des objets est un excellent moyen d'examiner les objets utilisés dans un patch plus complexe.

Plus d'informations sur les objets et les cordons de raccordement

Si vous regardez les objets dans ce patchwork, vous remarquerez que les **cordons de raccordement** sont un peu différents de ceux que nous avons vus précédemment. Tout d'abord, un certain nombre de cordons de raccordement ont des chemins complexes composés de plusieurs segments; en outre, certains d'entre eux sont de couleurs différentes. Voyons comment nous pouvons manipuler les cordons de raccordement eux-mêmes. Dans le menu **Options** de Max, cochez l'option **Cordons de connexion segmentés**. Cela nous permet de créer des cordons de raccordement qui sont courbés. Nous pouvons supprimer un patchcord en le sélectionnant et en appuyant sur la touche Suppr, ou en choisissant Supprimer dans le menu Patchcord. Supprimez l'un des cordons de raccordement reliant les boîtes de *message* à l'objet *gate* en haut à gauche du patch.

Maintenant, reconnectez la boîte de *messages* à *gate*, mais au lieu d'établir une connexion directe entre les deux, créez un cordon de raccordement **segmenté** en cliquant sur la sortie de la boîte de *messages*, puis en relâchant la souris; nous pouvons ajouter des segments en cliquant avec la souris n'importe où pendant que nous dessinons le patchcord; nous finissons la connexion en cliquant sur l'entrée de l'objet de destination (*gate*). L'utilisation de patchcords droits ou segmentés est en grande partie une question de style personnel, mais il est souvent possible de donner à un patch un aspect plus propre en utilisant des patchcords segmentés.

Max peut également nous aider à nettoyer nos cordons de raccordement. Cliquez sur le cordon de raccordement et appuyez sur Shift-Command-Y sur Mac, puis sur Ctrl-Shift-A sur votre PC (vous pouvez également sélectionner la fonction **Route Patch Cords** dans le menu **Arrangement** de Max). Max segmentera et réacheminera le patchcord pour éviter le chevauchement des objets. Un simple Command-Y permet d'obtenir un cordon avec un seul coude sur le chemin le plus court. Nous pouvons également sélectionner plusieurs cordons de raccordement en maintenant la touche Shift enfoncée tout en cliquant sur les cordons de raccordement (ou sur la touche Option tout en faisant glisser les cordons) - la séquence de commandes redirige alors tous les cordons de raccordement sélectionnés. Il n'y a aucune différence entre un cordon de patch régulier et un cordon de raccordement segmenté en termes de transmission des messages entre les objets.

modes

Enfin, examinons la capacité de Max à modifier la couleur des objets et des cordons de raccordement. Nous pouvons sélectionner un objet ou un cordon de patch (en cliquant dessus), puis sélectionner **Format** dans la barre d'outils supérieure. Cela affichera une nouvelle barre d'outils avec des icônes pour les éléments individuels des objets - l'arrière-plan, le texte, les bordures, tout ce qu'ils ont. La sélection de l'icône appropriée vous permettra de colorer (par exemple) le clignotement d'un bouton. Nous avons coloré les objets au fur et à mesure que nous les avons présentés dans les tutoriels; comme vous pouvez le constater, c'est un excellent moyen de focaliser l'attention sur des éléments importants de notre patch. (Notez qu'il existe une couleur dorée uniforme pour les objets d'un patch que vous serez invité à manipuler.)

En plus de colorier soigneusement chaque objet, vous pouvez enregistrer le jeu de couleurs en tant

que style. Le texte à gauche de la barre d'outils de formatage est un menu déroulant. Si un objet coloré est sélectionné, vous pouvez définir sa palette de couleurs en tant que style avec un nom distinct. Ces noms seront alors stockés dans le patch afin que vous puissiez facilement les appliquer à d'autres objets du même type. Vous pouvez même enregistrer vos styles dans une bibliothèque afin qu'ils soient disponibles dans tous vos patches. Enfin, plusieurs jeux de styles sont fournis si vous estimez que l'aspect par défaut est un peu austère.

*Travailler avec l'objet **gate***

Regardons maintenant ce que fait le patcheur. Dans notre patcheur de tutoriel, nous trouvons deux petits patches (à gauche) et un très grand patch. Le patch en haut à gauche est celui par lequel nous allons commencer.

Ce patch montre l'objet *gate* en action. L'objet *gate* est l'un des objets les plus importants utilisés pour le routage dynamique des messages dans Max. Nous allons commencer par cliquer sur le message intitulé **1** (la boîte de message du milieu), puis sur la valeur de la boîte de *nombres* associée à l'entrée droite de *gate*. Nous constatons que la valeur de la boîte de *nombres* est envoyée par la sortie gauche de *gate* (sortie **1**). Si nous cliquons sur le message intitulé **2** et modifions à nouveau la boîte de *nombres*, nous voyons que les modifications sont maintenant envoyées par la sortie droite de *gate* (ou à la sortie **2**).

Comme on peut le constater, le nombre envoyé dans l'entrée gauche détermine le routage du message reçu dans l'entrée droite; essentiellement, l'entrée gauche détermine quelle sortie est ouverte. Maintenant, cliquez sur le message étiqueté **0** et modifiez le *nombre* à droite. Le message ne sort jamais de *gate* - le message **0** a fermé toutes les sorties. Ceci est utile pour les situations où nous souhaitons supprimer des messages au sein de notre patch, par exemple lorsque nous avons besoin d'activer et de désactiver de manière sélective les nombres qui circulent d'une partie de notre patch à une autre.

En exercice, changez l'objet *gate* pour avoir un **4** pour un argument (c'est actuellement un **2**). Nous voyons que l'objet change - il a maintenant quatre sorties. L'argument pour *gate* détermine le nombre de sorties que possède l'objet. Si nous ajoutons quelques messages supplémentaires à l'entrée gauche (en ajoutant une option **3** et **4** pour prendre en charge les nouvelles sorties de *gate*) et en associant des boîtes de *nombres* aux nouvelles sorties, nous pouvons voir à quel point il est facile d'élargir la matrice de routage prise en charge par l'objet *gate*.

pack, unpack, prepend

La partie suivante du patch que nous allons examiner est juste en dessous de *gate*. Elle utilise les objets *Unpack* et *Pack* pour diviser et ré-assembler des listes de nombres, et l'objet *prepend* pour placer un sélecteur de message (un symbole) devant la liste. Les deux boîtes de *messages* contiennent des listes de deux nombres; lorsque nous cliquons dessus, nous voyons que les deux boîtes de *nombres* connectées à l'objet *unpack* affichent les nombres individuels de la liste. L'objet *unpack* a pris la liste entrante et l'a divisée en messages numériques individuels. Comme les listes sont souvent utilisées pour créer des messages complexes, nous utilisons souvent l'objet *unpack* pour décompresser ces listes dans ses composants individuels (appelés **atomes**).

Une fois que les boîtes de *nombres* ont affiché les messages, elles envoient leur contenu dans l'objet *pack* ci-dessous, ce qui inverse le processus en les combinant à nouveau en listes qui sont envoyées à l'objet *prepend* et affichées dans la console Max (via l'objet *print*). Remarque: un seul message est affiché dans la console Max, même si l'objet *pack* a reçu deux messages (un de chaque boîte de *nombres*), car l'objet *pack* (comme la plupart des autres objets de Max) ne produira un message que

lorsque l'entrée la plus à gauche reçoit un message. L'environnement d'édition des patches fournit un indice à cet égard: si nous déverrouillons le patch et que nous passons le curseur de la souris sur l'entrée droite, nous voyons qu'il affiche un cercle bleu, ce qui signifie qu'il s'agit d'une entrée **froide** (ne produisant pas de message). Survolez l'entrée gauche et voyez-la encerclée en rouge; c'est une entrée **chaude** qui provoquera une sortie lorsqu'il recevra un message. Nous pouvons le confirmer en tapant d'autres nombres dans les boîtes de *nombres* prises en sandwich entre le *unpack* et le *pack* - seul la boîte de *nombre* de gauche fera sortir une liste de l'objet *pack*.

Les arguments de *pack* et *unpack* déterminent à la fois le nombre d'entrées et de sorties, que l'objet aura, respectivement, mais également, le type de message pouvant être envoyé dans ces entrées. Dans notre cas, ces deux objets travaillent avec des listes à deux éléments composées de nombres entiers (signifiés par l'utilisation de **0** comme argument initial). Si la liste que nous composons requiert des nombres à virgule flottante, nous aurions pu utiliser **0.** (ou toute autre valeur initiale à virgule flottante) comme argument; de même, si un symbole (un mot) était nécessaire dans la liste, nous aurions pu utiliser la lettre **s** (ou n'importe quel symbole); les types de données peuvent être mélangés et assortis dans une liste.

Notez que l'objet *prepend* place le mot **thelist** devant la liste des nombres sortant de *unpack* - ce **symbole** peut être n'importe quel mot, mais dans de nombreux cas dans Max, nous utilisons souvent *prepend* pour placer un **sélecteur de message** au début d'une liste avant de l'envoyer dans un objet ayant une utilisation spéciale pour ce message.

L'objet lcd

Il est maintenant temps d'attaquer le grand patch à droite. Voyons ce qu'il fait en allumant l'objet *metro* en haut du patch en cliquant sur l'objet *toggle* sur 1 (X lumineux).

Le grand objet rectangulaire à droite du patch est un objet *lcd* (**affichage à cristaux liquides**): nous pouvons utiliser cet objet pour dessiner des pixels, des lignes, des formes, du texte et d'autres graphiques vectoriels bidimensionnels simples. Si vous faites glisser la souris dans cette zone tout en maintenant le bouton de la souris enfoncé, nous voyons qu'elle trace des lignes partout où vous allez. Relâchez le bouton de la souris et le dessin s'arrêtera.

Si vous maintenez la touche Maj, Option/Alt ou une combinaison des deux, l'objet *lcd* dessine des formes différentes sous la souris plutôt que des lignes. Il y a trois boîtes de message *colorées* au-dessus de *lcd* qui peuvent être utilisées pour modifier la couleur de premier plan du dessin (à l'aide du message **frgb**), et une autre boîte de *message* (à droite) qui enverra le message **clear** à l'écran en effaçant son écran.

L'une des sections clés de ce patch concerne la manipulation des touches Maj et Option/Alt. Cette opération est effectuée par l'objet *modifiers*, interrogé à l'aide du *metro* et indiquant le réglage actuel des différentes touches de modification de votre clavier. Dans ce cas, nous utilisons les sorties qui signalent les touches Maj et Option/Alt; nous **multiplions** l'état de la touche option par **2** et **l'ajoutons** à l'état de la touche shift. Le résultat est que shift seule nous donnera la valeur **1**, option/alt seule donnera la valeur **2** et la combinaison des deux nous donnera **3**. Si aucune touche de modification n'est enfoncée, la valeur sera **0**. Ce type de circuit de patch est parfait pour contrôler l'état de l'objet *gate* ci-dessous.

En plus de fournir une surface de dessin, l'objet *lcd* peut nous indiquer l'emplacement actuel de la souris d'une manière analogue à l'objet *mousestate*, uniquement par rapport à la position de *lcd* dans le patch. L'envoi du message idle **1** à l'objet nous permet d'obtenir les coordonnées de la souris, que la souris soit cliquée ou non dans l'écran *lcd* (nous avons activé la fonction idle mousing

lorsque nous avons activé la boîte du *toggle*, ci-dessus).

Ces coordonnées sont fournies par la deuxième sortie de l'objet *lcd* - nous pouvons voir les valeurs (une liste de deux nombres représentant les coordonnées X et Y de l'objet *lcd*) dans les boîtes de nombres situées en bas, après qu'elles aient été séparées par l'objet *unpack*. Cette liste est également envoyée à l'entrée droite de l'objet *gate*. Nous avons coloré ce cordon de couleur rouge pour qu'il soit facile à voir.

Les objets nommés **p** sont des objets *patcher*. Ceux-ci contiennent des sub-patches qui génèrent les formes affichées sur *lcd*. En acheminant les coordonnées de la souris vers différents patchers, l'objet *gate* contrôle maintenant le type de dessin qui va se produire. Si aucune touche de modification n'est enfoncée, la *gate* reçoit un **0** et aucune des routines de dessin n'est utilisée, bien que nous puissions toujours cliquer-glisser à l'intérieur de l'écran *lcd* pour dessiner des lignes. Si *shift*, *option/alt*, ou les deux, sont enfoncées, la *gate* dirige l'emplacement de la souris vers le subpatch approprié.

- Double-cliquez sur un objet *patcher* (**p**) pour voir ce qu'il contient. Notez que les cordons de connexion menant à l'entrée et à la sortie du sub-patch sont terminés par des objets avec des flèches. Ce sont des objets d'*entrée* (en haut, apportant des messages) et de *sortie* (en bas, envoyant des messages)

Dans chacun de ces sub-patches, les emplacements X et Y sont passés par des objets *unpack* pour obtenir les coordonnées individuellement. Elles sont ensuite utilisées pour créer une forme correctement dimensionnée. Dans le cas des **rectangles** (*shift*), une valeur de **10** est soustraite et ajoutée aux deux coordonnées et envoyée à un objet *pack* dans l'ordre suivant:

X-10 Y-10 X + 10 Y + 10

Cette liste de quatre valeurs est ensuite envoyée à un *prepend*, ce qui place un symbole au début de la liste. L'objet *lcd* supporte un certain nombre de commandes de dessin qui font différentes choses; dans ce cas, nous demandons à notre écran *lcd* de dessiner des cadres rectangulaires en lui fournissant un message **framerect** suivi de quatre arguments: les limites gauche, supérieure, droite et inférieure du rectangle que nous souhaitons dessiner.

En regardant les deux autres morceaux de code de dessin, nous pouvons voir qu'ils fonctionnent de manière similaire. La sortie intermédiaire de l'objet *gate* (répondant à la touche *option/alt*) dessine des images de forme ovale (le message **frameoval**) d'une taille de 10 x10 pixels (remarquez les objets + et -). La troisième variante de dessin (lorsque les touches *Maj* et *Option/alt* sont toutes deux maintenues enfoncées) dessine des rectangles aux coins arrondis (**paintroundrect**). En plus des coordonnées du rectangle, deux boîtes de *nombres* supplémentaires sont envoyées dans l'objet *pack*: elles contrôlent le degré d'arrondi des angles.

Lorsque les coordonnées sont créées et déterminées (à l'aide des objets mathématiques et de *pack*) et qu'un sélecteur de message leur est attribué pour déterminer la routine de dessin (à l'aide de l'objet *prepend*), le message est renvoyé dans l'objet *lcd*. Pour voir facilement cet acheminement des messages, nous avons coloré les cordons de connexion de dessin en bleu. Les boîtes de *message* de **frgb** sont également colorées pour correspondre à la couleur avec laquelle elles indiqueront à l'écran *lcd* de dessiner. Ce sont des exemples d'utilisation de la coloration d'objets et de cordons de patch pour rendre plus évidente la fonction d'un patch.

Résumé

Plusieurs sujets différents (mais importants) ont été abordés dans ce tutoriel. Nous avons vu comment l'objet *gate* peut être utilisé pour activer le passage de messages vers des sections de votre patch. Nous avons également vu comment utiliser les objets *Unpack*, *Pack* et *Prepend* pour créer et analyser des messages complexes. Nous avons également vu comment colorer et manipuler des cordons de patch et des objets pour rendre plus facilement visibles les patches et le routage des messages. Enfin, nous avons vu comment utiliser l'objet *lcd* pour réaliser des dessins simples.