

## 31-Cellblock

### introduction

Ce tutoriel se concentre sur l'objet *jit.cellblock* (communément appelé «cellblock»). L'objet *jit.cellblock* fournit une interface visuelle à une matrice bidimensionnelle de données, ou chaque point de données (ou «cellule») peut contenir n'importe quel type de données utilisés dans Max. Ces données peuvent être modifiées et éditées et sélectionnées par l'utilisateur ou par le patch.

L'objet *jit.cellblock* répond à un besoin particulier de Max: un affichage peu encombrant des valeurs Max, avec la possibilité de les sélectionner ou de les modifier visuellement dans le patch. Cet objet, comme l'objet *lcd*, supporte un très grand nombre de messages et de paramètres d'attributs. Dans ce didacticiel, nous couvrirons certains des plus utiles, mais de nombreuses autres options sont disponibles pour une utilisation avancée.

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

### Messages structurels et configuration

Jetez un coup d'oeil au patcheur de ce tutoriel. Vous verrez un affichage basé sur une grille, avec trois ensembles de boîtes de *messages*. La grille au milieu du patcheur (qui ressemble à une feuille de calcul) est l'objet *jit.cellblock*; il fournit un affichage bidimensionnel de données stockées dans une structure interne.

De nombreux paramètres structurels du *jit.cellblock* sont disponibles pour le contrôle par programme. Par exemple, deux boîtes de *nombres* sont connectées aux messages **rows** et **cols**. Ces boîtes de *nombres* peuvent être utilisées pour modifier manuellement le nombre de lignes et de colonnes stockées dans le bloc *jit.cellblock*. Si vous ajoutez plus de lignes ou de colonnes que ce qui peut être affiché dans la zone disponible, des barres de défilement apparaissent automatiquement.

Un autre type de structure peut être impliqué en modifiant la couleur de la première colonne ou ligne avec l'objet. Cela permet d'indiquer clairement que cette ligne est un "index" ou un "identifiant" pour le reste de la ligne. En utilisant les messages **rowhead** et **colhead**, vous pouvez modifier l'utilisation perçue de cette première valeur. Il est important de noter que, si la zone d'en-tête subit un changement visuel, l'utilisation des données ou la possibilité de les ajouter ou de les modifier ne sont pas affectées.

Il est possible de modifier d'autres attributs visuels qui n'ont aucun effet sur les données sous-jacentes. Un exemple est l'attribut **grid**. Lorsque la grille est activée, les données s'affichent comme dans une feuille de calcul. Cependant, lorsqu'elle est désactivée, l'affichage des valeurs n'est pas délimité par la grille dessinée. Cette option est particulièrement pratique si vous souhaitez une disposition de grille moins distincte qui utilise du texte ou des couleurs de fond pour distinguer les sous-ensembles de données.

D'autres paramètres structurels peuvent être manipulés, et l'ouverture de l'inspecteur d'objets permet d'accéder à la plupart d'entre eux. Si nous ouvrons l'inspecteur d'objets pour l'objet *jit.cellblock* dans le patch du didacticiel et sélectionnons la section **Structure**, nous voyons des paramètres d'attribut qui correspondent aux messages basés sur les patches que nous venons d'utiliser. De plus, il existe des paramètres pour les tailles de colonnes et de lignes par défaut: la largeur et la hauteur de ces éléments si elles ne sont pas spécifiquement définies. Si vous ajustez ces paramètres, les cellules du *jit.cellblock* seront changer dynamiquement pour s'adapter.

D'autres paramètres visuels sont disponibles dans la section apparence de l'inspecteur. Les couleurs des différents éléments du *jit.cellblock* peuvent être définies à l'aide du sélecteur de couleur; cliquez sur la couleur que vous souhaitez modifier et un sélecteur de couleurs standard sera affiché. Sélectionnez la **couleur d'arrière-plan par défaut** et changez la couleur en rouge vif. Vous remarquerez que l'arrière-plan de toutes les cellules sera modifié, mais que les autres éléments (tels que la couleur de la grille ou la cellule actuellement sélectionnée) resteront inchangés.

Familiarisez-vous avec les différents paramètres situés à gauche du patch (intitulés **1.Structure** et **Apparence**), ainsi que certaines des valeurs de l'inspecteur pour l'objet *jit.cellblock*.

## Stockage et récupération de valeurs

Au-delà de la configuration de la structure de l'objet *jit.cellblock*, nous devons sélectionner des cellules, y stocker des données et les récupérer. La façon la plus simple de sélectionner une cellule, consiste à cliquer dessus, un peu comme vous le feriez avec une feuille de calcul. Vous remarquerez que la sélection d'une nouvelle cellule produit les coordonnées (dans l'ordre colonne / ligne) de la sortie gauche de l'objet (celles-ci sont *imprimées* dans la Max console). Etant donné que vous souhaitez que le programme contrôle la sélection en cours, l'objet *jit.cellblock* répondra également à un message de **sélection**, avec des arguments qui choisissent la colonne et la ligne à sélectionner. Vous le trouverez dans la section **2.Contenu**. En utilisant les deux boîtes de *nombre* attachées au message de sélection, vous serez en mesure de contrôler la sélection actuelle du *jit.cellblock*. Notez que le numéro de ligne et de colonne sont tous deux basés sur zéro, ce qui signifie que la première ligne et la première colonne sont sélectionnées en envoyant le message **select 0 0**.

La ou les cellules actuellement sélectionnées peuvent être appelées «actuelles» dans certains messages. La boîte de *nombres* suivante est acheminée par un message **set current**; celui-ci prendra la valeur entrante et changera la valeur de la cellule courante en ce nombre entier. Testez-le en sélectionnant plusieurs cellules différentes et en modifiant la valeur. Maintenant, cliquez sur n'importe quelle cellule contenant une valeur: vous verrez que la sortie de l'objet a changé, avec la Max console affichant non seulement les numéros de colonne et de ligne, mais aussi les données contenues dans la cellule.

Il n'est pas nécessaire de sélectionner une cellule pour modifier les données qu'elle contient. La boîte de *message* suivante utilise la syntaxe **set x y value ...** pour définir le contenu de l'objet. Les deux nombres entiers qui suivent le message **set** représentent la colonne et la ligne qui seront modifiées. Toutes les données suivant ces coordonnées seront stockées. Si la donnée est un entier ou un nombre à virgule flottante, cette seule entrée sera stockée. Les symboles et les listes peuvent également être stockés comme valeurs pour une cellule. En cliquant sur la troisième boîte de message, la cellule **2 2** sera forcée de contenir le texte **Me voici !**. Bien entendu, nous devons également pouvoir être en mesure de supprimer les données, ce qui est la fonction du message **clear**. La boîte de *message* suivante génère le message **clear 2 2**, qui effacera le texte que nous venons de définir.

L'objet *jit.cellblock* est utile en partie parce qu'il fournit un large éventail d'options utilisateur pour sélectionner et manipuler les données dans une cellule. Bien sûr, si nous avons pris la peine de définir les valeurs des cellules, nous ne voudrions pas que l'utilisateur (ou le programme, en fait) modifie ces valeurs. Définir l'attribut **readonly** sur **1** empêchera la modification des valeurs. Lorsque cet attribut est défini, vous pouvez utiliser le *jit.cellblock* sans vous soucier de la modification des valeurs.

## Options de sélection et de sortie

Afin d'être aussi flexible que possible, *jit.cellblock* dispose de plusieurs modes de sélection et de sortie. Le mode de sélection (défini avec l'attribut *selmode*) sélectionne par défaut une seule cellule (mode dit "one cell select"). Si nous regardons le *umenu* à gauche de la section **3. Sélection et sortie**, nous constaterons qu'il existe six options différentes pour la sélection de cellules.

La première option (option **0**, "aucune sélection") empêche toute sélection de cellule. Cela peut être utile si vous souhaitez que l'objet *jit.cellblock* agisse comme un grand ensemble d'étiquettes. L'option **1**, "sélection de cellule unique", est le gestionnaire de sélection par défaut, où une seule cellule peut être sélectionnée à la fois. Les deux options suivantes (**2** et **3**) vous permettent de sélectionner un grand nombre de cellules en un seul clic. L'option 2 ("sélection de colonne") sélectionne toutes les cellules de la colonne que vous sélectionnez. L'option 3 ("row select") fait de même pour les valeurs de ligne.

La cinquième option (option **4**, «sélection d'en-tête») est plus complexe. Si les en-têtes de colonne ou de ligne sont activés et que vous cliquez sur une cellule d'en-tête, la colonne ou la ligne entière sera sélectionnée. Sinon, seule une cellule sera sélectionnée. Ceci ressemble beaucoup au fonctionnement d'une feuille de calcul, où les en-têtes ont des règles de sélection spéciales différentes des cellules standard.

La dernière option (option **5**, «édition sur place») offre un moyen facile d'entrer des données dans le fichier *jit.cellblock*. Double-cliquer sur n'importe quelle cellule vous permettra de modifier ou d'ajouter des informations dans la cellule sélectionnée. Cela peut être plus pratique que d'utiliser les messages *select* et *set*, bien que cela ne puisse pas être contrôlé avec votre patch.

Étant donné que plusieurs de ces modes sélectionnent plusieurs cellules, la question suivante se pose : «Dans quel format les données seront-elles présentées ?» L'objet *jit.cellblock* fournit trois modes de sortie comme réponse - ceux-ci sont définis avec l'attribut *outmode*. Le mode **0**, ou mode «sortie individuelle», enverra chaque cellule sélectionnée dans son propre message et émettra un message pour les cellules qui peuvent ne pas avoir de valeur de données. Le mode **1**, ou «sortie en une seule liste», combine toutes les valeurs de cellules non vides en une seule liste et les enverra dans un seul message. Le mode **2**, ou «sortie sous forme d'un symbole», combinera toutes les valeurs de cellules non vides dans une liste, puis transformera la liste en un seul symbole de texte. Ce mode est particulièrement utile si vous avez rempli votre *jit.cellblock* avec des valeurs de textes et que souhaitez les combiner dans des phrases plus grandes.

Pour voir le mode de sortie en action, cliquez sur le *button* associé au subpatch **loader**. Cela va formater et charger l'objet *jit.cellblock* avec des nombres de **0** à **99**. Le *jit.cellblock* est également basculé en mode «sélection de ligne», nous allons donc sélectionner de nombreuses cellules à chaque clic. Cliquez sur une cellule pour sélectionner une ligne. vous remarquerez que 10 messages apparaissent dans la Max Console de manière séquentielle, un pour chaque cellule sélectionnée. Maintenant, changez le mode de sortie en "sortie comme une liste", et cliquez sur la sixième ligne. La Max Console affiche maintenant un résultat sur une seule ligne, ce qui signifie qu'un seul message a été généré. Les deux premiers nombres sont la colonne (non pertinente dans ce cas) et la ligne (5, puisque *jit.cellblock* est basé sur zéro), puis une liste de toutes les valeurs de cette ligne.

Si nous changeons le mode de sortie en «sortie sous forme d'un symbole», les résultats dans la console max ne semblent pas différents. Cependant, si vous regardez la sortie de l'objet *zl len* (traité en détail dans un autre tutoriel) connecté à la sortie de l'objet *jit.cellblock*, vous verrez que notre message n'a que **3** éléments, plutôt que les **12** éléments que nous avons vus dans mode précédent.

## Résumé

L'objet *jit.cellblock* est un outil robuste permettant de traiter des données 2D de tout type et de présenter des données dans un format de type grille. Nous n'avons abordé que quelques-unes des commandes les plus utiles; pour plus d'informations, consultez les fichiers d'aide et la documentation de référence sur cet objet.