

13: Séquençage vidéo

introduction

Dans ce tutoriel, nous allons travailler avec quelques nouveaux objets: les objets *rslider*, *umenu* et *loadbang*. L'objet *rslider* nous permet de sélectionner une plage de nombres à l'aide d'un seul élément d'interface utilisateur. L'objet *umenu* nous permet de choisir un élément dans un menu déroulant d'un patcheur. L'objet *loadbang* est un objet de configuration. Il nous permet de déclencher des messages lorsqu'un patcheur est chargé pour la première fois. Tous ces éléments sont importants pour les patches plus complexes - et nous les utiliserons dans le cadre de la création de quelques programmes vidéo intéressants.

Il est important de fournir des sélections utilisateur une fois que votre patch devient compliqué - ou si vous souhaitez partager votre patch avec d'autres. L'objet *rslider* est idéal pour la sélection de points de boucle (puisque'il vous permet de sélectionner les points de début et de fin d'un seul glissement de souris) et que l'objet *umenu* est parfait pour sélectionner parmi une liste de fichiers ou d'options d'affichage. Enfin, *loadbang* nous donne la possibilité de définir les paramètres d'objet lorsqu'un patch est chargé, ce qui nous permet de créer des patches qui ne nécessitent qu'un minimum de configuration pour être utilisés.

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

Quelques nouveaux objets: *rslider*, *loadbang*, *umenu*

Jetez un coup d'œil au patch du tutoriel. Ce patch contient quelques patches simples à gauche et deux patches plus grands à droite. Commençons par les patches de gauche, où nous découvrons nos trois nouveaux objets.

Le patch le plus à droite est un patch d'exemple pour l'objet *rslider*. Cliquez sur la surface et vous verrez que l'essuie-glace (le cas échéant) disparaîtra. Cependant, si vous cliquez avec la souris puis faites glisser vers le haut ou le bas sur la surface, vous verrez que vous pouvez en fait sélectionner une plage de valeurs - reflétée dans les deux sorties de l'objet. Si vous faites un shift-clic, vous pouvez ajuster l'extrémité la plus proche de la souris. Le nom *rslider* fait référence au fait qu'il s'agit d'un « curseur de plage », capable de sélectionner une plage de valeurs en un seul mouvement de souris. La plage réelle couverte par le *rslider* peut être définie avec un simple message (la valeur par défaut est comprise entre **0** et **127**), comme nous le verrons plus loin dans le didacticiel.

L'objet suivant est assez simple, mais très important. C'est l'objet *loadbang*, et il envoie un **bang** à chaque fois que le fichier patcheur est chargé (d'où son nom). Ceci est le plus souvent utilisé pour définir des valeurs dans une interface utilisateur - dans ce cas, nous chargeons la boîte de *nombres* avec **55** (normalement, la valeur par défaut est **0**) chaque fois que nous ouvrons le patch. Nous pouvons également forcer le *loadbang* à se déclencher en double-cliquant dessus. Si nous changeons la boîte de **nombres** connectée par un autre nombre, le fait de double-cliquer sur le *loadbang* provoquera l'envoi d'un message **bang** dans la boîte de **messages** (contenant le message **55**), et la boîte de nombres sera de nouveau mise à jour à sa valeur précédente.

Loadbang ne produit pas de bang lorsqu'il est créé ou copié. Il est donc facile de désynchroniser les nouvelles tâches avec ce qui se passera lorsqu'un patcheur sera chargé. Assurez-vous de leur donner un double-clic avant de tester un patch. Mieux encore, enregistrez, fermez et rouvrez le patch pour confirmer que *loadbang* fait bien son travail. Si vous devez dupliquer automatiquement l'action d'un *loadbang*, il répondra à un bang dans son entrée en envoyant un bang.

Enfin, le *umenu* est affiché en bas des trois petits patches. Si vous cliquez sur *umenu*, un ensemble d'options apparaît sous forme de menu contextuel. La sélection de l'une d'entre elles produira deux formes de sortie: la sortie de **gauche** enverra le numéro d'index de la sélection, en commençant à **0**; et la sortie du **milieu** émettra un symbole contenant le texte de l'option de menu.

Les éléments disponibles dans un objet *umenu* peuvent être définis de différentes manières. Dans la section suivante, nous verrons comment le remplir automatiquement. Cependant, commencez par **déverrouiller** le patch et ouvrez l'**inspecteur** pour l'objet *umenu* dans la barre d'outils de droite. Dans l'inspecteur, vous verrez une section intitulée "Items"; Dans cette section, nous pouvons voir la liste des éléments contenus dans le *umenu* (**petit, moyen**, etc.). Cliquez sur le bouton "Editer" à côté de la liste et une fenêtre s'ouvrira qui nous permettra de taper tout ce que vous voulez. Les éléments de l'*umenu* doivent être séparés par des virgules. Ajoutez à la liste (ou créez-en une entièrement nouvelle), cliquez sur **OK** et fermez l'inspecteur. Vous devriez voir l'*umenu* changer pour refléter les nouveaux éléments que nous venons d'ajouter.

Chargement de *umenu* avec les informations sur les fichiers

Le premier grand patch porte le libellé 1 et comporte deux sections intéressantes: la zone supérieure sert à sélectionner les fichiers vidéo d'un dossier contenant les films livrés avec Max, tandis que la zone inférieure sert à boucler la lecture de ces films au sein d'un objet *imovie*. Commençons par regarder la partie supérieure.

Nous avons ici un système de sélection de fichiers plutôt complet. Un objet *loadbang* est utilisé pour déclencher une boîte de messages qui envoie le message **préfixe./patches/media** à l'objet *umenu*. Si nous examinons le matériau de référence pour l'objet *umenu*, nous constatons que le **préfixe** est un attribut qui définit un préfixe pour l'ensemble du menu qui est ajouté au début de tout élément sélectionné. Il a également une deuxième utilité: il indique à l'*umenu* où charger les fichiers lorsqu'on lui demande de se remplir (ou, dans notre cas, de se **remplir automatiquement**). L'attribut autopopulate de l'objet *umenu* est défini dans l'inspecteur. Ainsi, lorsque le message **préfixe** est reçu, l'objet *umenu* est chargé avec les noms de tous les fichiers du dossier indiqués par le message **préfixe**.

En conséquence, nous avons un *umenu* chargé avec les noms de fichiers d'un dossier appelé "media" à l'intérieur d'un dossier appelé "patches" (le dossier "patches" est en fait trouvé dans le paquet d'application Max. Sinon, cette forme ferait référence aux dossiers relatifs au dossier où Max est installé sur votre système. La plupart du temps, vos chemins seront relatifs à ~utilisateurs.) Lorsque nous en sélectionnons un, trois choses se produisent, grâce à l'objet *trigger* connecté à la sortie centrale de l'*umenu*. Tout d'abord, le message **dispose** est envoyé à l'objet *imovie*; cela vide l'objet *imovie* de tout fichier vidéo qui y est actuellement chargé. Ensuite, le symbole (le **s** dans *trigger*) contenant le nom du fichier précédé du préfixe de dossier **./patches/media** est envoyé. Le message **read** est placé devant (via l'objet *prepend*) puis envoyé à l'*imovie*. Troisièmement, l'objet *trigger* envoie un **bang** via un objet *delay*, ce qui retarde le message de **100** millisecondes; cela donne à l'objet *imovie* le temps de lire sans le film. Une fois le délai expiré, la boîte de message au-dessous est déclenchée. Elle contient quelques messages d'initialisation dans *imovie*: **longueur** (pour demander la longueur du film), **boucle 1** (pour indiquer au film de tourner en boucle) et **début** (pour commencer la lecture). Dans ce cas, bien que nous ayons une boîte de *nombres* connectée à la sortie gauche, nous ne faisons rien avec l'index de l'objet *umenu* - tout est déclenché par la sortie centrale (texte). Notez que l'objet *trigger* est extrêmement utile dans ce cas, car il nous permet d'établir une série complexe d'étapes dans l'ordre correct: disposer de la vidéo précédente, lire une nouvelle vidéo, déterminer sa longueur, lui dire de boucler et la démarrer.

Sélectionnez une vidéo dans *umenu* et notez qu'elle se chargera rapidement et commencera à être lue en boucle à la fin du clip. Si vous sélectionnez un autre clip, le précédent sera supprimé et le nouveau commencera à tourner en boucle.

Utilisation de *rslider* pour définir des points de boucle

En plus de charger et de démarrer le film, ce message **length**, **boucle 1**, **start** a eu une autre conséquence: il a forcé la longueur du film à sortir de la sortie gauche de l'objet *imovie* et a déclenché une nouvelle chaîne d'événements. Ce message de **length** est utilisé pour définir la taille (c'est-à-dire la **plage**) d'un *rslider* et définir l'objet avec la plage complète sélectionnée (c'est ce que fait le message **0 \$ 1**). Nous pouvons facilement modifier le contenu du *rslider* en cliquant et en glissant à l'intérieur de celui-ci, modifiant ainsi les points de boucle utilisés par *imovie*. Pour mieux voir cela en action, sélectionnez le film "countdown.mov" dans le *umenu* en haut - c'est le film familier qui compte à rebours de **10** à **1**.

Maintenant, si nous cliquons et faisons glisser le *rslider* au bas du patch, vous pouvez voir que nous changeons les points de début et de fin de la boucle. La sortie de l'objet *rslider* est envoyée à un objet *pack*, puis utilisée pour créer un message de **loopset**, qui modifie les "points d'entrée et de sortie" en boucle honorés par *imovie*. En outre, il rembobine le film au début de la boucle (le message **\$ 1**). Cette section du patch montre une utilisation très courante de l'objet *rslider*; il est initialisé avec la longueur d'un fichier multimédia, puis est utilisé pour les positions de début et de fin de lecture.

Mise en place: *loadbang* et *screensize*

Enfin, nous allons voir comment *loadbang* (ainsi que quelques autres messages) peut être utilisé pour configurer complètement un patch pour l'utilisateur lors de sa première ouverture. Ce patch (intitulé **2**, à droite) prend la position actuelle de la souris et l'utilise pour décider où placer le film dans une zone d'affichage de 320 x 240 pixels. Le *mousestate* est interrogé et un message **rect** place le film en cours de lecture, sous forme réduite, à l'emplacement spécifié dans l'objet *imovie*.

Pour commencer, un message *loadbang* est utilisé pour charger le fichier vidéo «ball.mov» lors du premier chargement du patch. De cette façon, vous (ou tout utilisateur) n'avez pas à rechercher les fichiers par vous-même. Deuxièmement, un autre *loadbang* est utilisé en haut à droite du patch; il envoie un message **bang** à l'objet *screensize*. L'objet *screensize* génère une liste de quatre coordonnées représentant les coordonnées de votre écran principal. Les deux derniers éléments de cette liste vont, à toutes fins utiles, nous donner la largeur et la hauteur, en pixels, de notre écran (par exemple, **1024** et **768**). Comme nous nous en souvenons, l'objet *mousestate* affiche la position de la souris dans cette plage en partant de **0**; par conséquent, si nous soustrayons **1** de la taille de l'écran, nous pouvons redimensionner correctement la sortie du *mousestate* dans la plage **0**. à **1**. c'est une astuce courante lorsqu'on travaille avec les coordonnées de la souris, car nous utilisons rarement leurs valeurs *réelles* en pixels; le plus souvent, nous souhaitons simplement les utiliser pour contrôler une plage de nombres entièrement différente.

Une fois que l'objet *scale* a son entrée maximale définie par l'objet *screensize*, la position actuelle de la souris donnera toujours des valeurs comprises entre 0.0 et 1.0. Ces valeurs sont ensuite multipliées par **240** et **180** pour obtenir l'emplacement supérieur gauche de l'écran pour le placement de la vidéo dans l'objet *imovie*. La partie **80 60** du message **rect** définit la **largeur** et la **hauteur** du rectangle au sein de *imovie* - ainsi, pour une fenêtre de **320** sur **240**, nos coordonnées d'entrée maximales doivent être **240** et **180**.

Activez l'objet *metro* avec le *toggle* pour commencer à interroger la souris, puis cliquez sur le message **start**. Déplacez votre souris autour de l'écran. Vous devriez voir une petite version du film de basket-ball suivre l'emplacement de votre curseur actuel et celui-ci sera toujours mis à l'échelle correctement, quel que soit l'affichage sur lequel vous l'exécutez.

Résumé

Dans ce didacticiel, nous avons découvert trois nouveaux objets très importants: *rslider* pour la sélection de plage, *umenu* pour la création et la sélection d'une liste d'items (ou noms de fichiers) et *loadbang* pour la configuration de l'état initial. De plus, nous avons vu comment l'objet *screenize* peut être utilisé pour obtenir des informations sur la taille de votre écran. Nous les avons illustrés en créant quelques systèmes complexes de lecture vidéo contrôlant la longueur de la boucle et le positionnement de l'affichage. Ces objets seront parmi les outils les plus importants que vous utiliserez dans les futurs patches.