

36-bpatchers

introduction

Ce tutoriel examine un objet utile pour incorporer des subpatchers dans un patch de façon que leur interface soit visible. L'objet *bpatcher* est superficiellement similaire à la création de n'importe quelle abstraction: nous créons un fichier patcher (avec les entrées et les sorties, le cas échéant) et nous le sauvegardons sur le disque. Cependant, plutôt que de l'instancier avec une boîte objet, nous utilisons l'objet *bpatcher* pour «l'incorporer» visiblement dans un autre patcher. Cela nous permet de construire une bibliothèque d'éléments d'interface - des éléments de logique de patcheur qui montrent une interface que nous pouvons réutiliser plusieurs fois.

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

Vue d'ensemble du patcher du tutoriel

Jetez un coup d'œil au patcheur du tutoriel. La fenêtre relativement clairsemée contient un *toggle* intitulé "Démarrer", trois objets *button* ("dessiner", "voir" et "effacer"), ainsi que trois zones d'interface d'apparence identique, chacune avec un "*button* clear".

Cliquez sur le *toggle* intitulé "Démarrer". Déplacez la souris sur la zone d'interface de gauche et dessinez dans le rectangle bleu qui contient la petite boule jaune. Ceci est un objet *jsui* chargé d'une interface utilisateur qui présente une fonction X-Y pour suivre la souris lorsqu'elle est cliquée. Au fur et à mesure que vous dessinez en glissant dans le *jsui*, le geste de votre dessin sera tracé en rouge dans l'objet *lcd* en dessous. La bordure de l'objet *lcd* étant désactivée, vous ne verrez rien tant que vous n'aurez pas dessiné quelque chose dans le *jsui*. Si vous effacez l'écran *lcd* (avec le *button* «erase»), vous devriez voir votre geste dessiner à nouveau.

Allez aux deuxième et troisième objets *jsui* et "dessinez" dedans avec la souris. Remarquez qu'ils génèrent des tracés dans les objets *lcd* en vert et en bleu, respectivement. Appuyez sur le *button* «erase» pour effacer l'écran *lcd* et les voir se redessiner.

Augmentez les objets *sliders* à côté des objets *jsui* et vous verrez que les tracés deviennent bruyants - les gestes enregistrés par le *jsui* sont maintenant un peu aléatoires pour ajouter plus de texture au dessin. Cliquez sur l'un des objets *button* portant la mention "clear" pour effacer le geste capturé. Vous pouvez ensuite redessiner un nouveau geste dans le *jsui* approprié.

Avec quelques gestes chargés, cliquez sur le *button* "see" en haut du patcheur. Notez que les trois zones d'interface ont soudainement un aspect différent: un nouvel ensemble d'objets *jsui* apparaît qui reproduit les gestes tels que vous les avez saisis et les contrôles associées ont disparu. Où sont passés le *jsui*, le *slider* et le *button* "clear" de chaque interface ?

L'objet *bpatcher*

Déverrouiller le patcheur du tutoriel. Notez qu'un certain nombre d'objets cachés apparaissent; de plus, des boîtes bleu clair apparaissent autour des trois zones d'interface. Les zones encadrées en bleu sont en fait des objets *bpatcher* contenant un patcheur d'abstraction enregistré sur le disque. Ces fichiers de patcheur ont des parties de leur interface montrées à l'intérieur du patch principal du tutoriel. Le concept est qu'un patch contenu dans le *bpatcher* se trouve derrière le patch principal et apparaît lors de son ouverture.

Cliquez sur l'une des trois zones d'interface et notez que, plutôt que de pouvoir modifier les objets qu'il contient, vous pouvez simplement déplacer et redimensionner la zone du *bpatcher* (zone bleue). Tout en maintenant la touche Ctrl-click enfoncée (Mac) ou click droit (Windows) sur l'une des boîtes bleues, vous obtiendrez son menu contextuel. Dans le menu, rendez-vous au sous-menu **Objet** et sélectionnez "Open Original doodlepatch.maxpat". La copie principale des trois zones d'interface s'ouvre. Jetons un coup d'oeil à ce patcher.

Déverrouillez l'abstraction "doodlepatch.maxpat". Remarquez que tout ce que nous voyons, ce sont les éléments d'interface du patcher, que le patcher soit déverrouillé ou non. Cela est dû au fait que le patch est en **mode présentation**: vous pouvez configurer n'importe quel patcheur pour qu'il s'ouvre en mode présentation en cochant la case "Ouvrir en présentation" dans l'inspecteur des patcheurs. Dans le menu **Affichage**, décochez l'option **Présentation** et le patcheur repassera sur sa vue d'origine et un ensemble de logique de patcheur apparaîtra.

Jetez un coup d'oeil au code du patcheur. Il utilise un objet *coll* ainsi qu'une certaine logique de soutien pour prendre les données de dessin de l'objet *jsui* et les enregistrer (il s'agit d'une forme de capture de geste, comme nous l'avons vu dans le "Didacticiel de données 3"). Le deuxième *jsui* (à droite) réanime immédiatement les données du *coll* en réponse aux messages **bang** venant de l'entrée et entraînant un objet *counter*.

Compenser un *bpatcher*

Sans fermer le fichier "doodlepatch.maxpat", revenez au patcheur du didacticiel et observez les objets "draw" et "see" lorsque le patcheur est déverrouillé. Remarquez qu'ils déclenchent des boîtes de messages contenant les *messages* **offset -60 0** et **offset -360 0**, respectivement, envoyés dans la deuxième entrée des objets *bpatcher*.

Regardez dans le patch *bpatcher* et voyez à quoi la deuxième entrée est connectée. Notez qu'elle communique avec un objet *thispatcher*. L'objet *thispatcher* fournit une fonctionnalité permettant de contrôler l'apparence d'un fichier patcher, notamment sa taille, sa position à l'écran et la visibilité ou non de la barre de titre. Lorsqu'il est utilisé dans un fichier *bpatcher*, l'objet *thispatcher* peut faire défiler l'interface de l'objet *bpatcher* en fonction des coordonnées fournies par le message **offset**. Cela permet à notre patcheur de tutoriel principal de passer de deux points de vue différents sur le même fichier *bpatcher* - dans ce cas, la vue "draw" et la vue "see".

Arguments pour les objets *bpatcher*

Examinez la logique du patcheur attachée à la sortie du fichier *bpatcher*. Il y a un objet *append* avec trois arguments d'abstraction **en forme de dièse** (# 1 # 2 # 3). Ces arguments devraient normalement être fournis dans la boîte de l'objet si nous devons charger "doodlepatch.maxpat" comme une abstraction normale dans notre patcheur. Afin de définir ces arguments dans un objet *bpatcher*, nous devons utiliser **l'inspecteur** de l'objet.

Fermez le fichier d'abstraction *bpatcher* et revenez au patcheur du didacticiel. Cliquez sur l'un des objets *bpatcher* et ouvrez son inspecteur. Notez qu'en bas, vous pouvez trouver le nom du **fichier du patcheur** chargé par l'objet. Juste en dessous se trouve ce champ se trouve le champ **Argument (s)**, qui contient trois chiffres. Ces chiffres sont substitués aux arguments du patcheur pour chaque *bpatcher* indépendamment, ce qui nous permet de différencier plusieurs copies de ceux-ci dans notre patch.

Les arguments de notre *bpatcher* dans ce tutoriel contrôlent les trois couleurs dessinées par le système de capture des gestes dans le *lcd* (en bref, ils fournissent les trois derniers chiffres des

commandes **frameoval** envoyées au *lcd*). Ouvrez l'inspecteur pour chaque *bpatcher* et changez ces valeurs en une valeur de couleur différente sous forme RGB (**0-255**). Lorsque vous enregistrez de nouveaux gestes, vous les verrez apparaître dans le *lcd* en utilisant les couleurs que vous avez sélectionnées.

Résumé

L'objet *bpatcher* fournit un moyen extrêmement utile d'intégrer des interfaces visuelles dans un projet plus vaste. Une interface peut être conçue (avec toute la logique de support requise), sauvegardée et affichée dans un autre patch. Le message **offset** d'un objet *thispatcher* dans un *bpatcher* nous permet de "faire défiler" le rectangle de visualisation de l'objet pour révéler différentes parties de l'interface. Ceci, combiné à l'utilisation du mode de Présentation, nous permet de personnaliser davantage l'interface. Comme pour les autres types d'abstractions, les patcheurs conçus pour être utilisés avec l'objet *bpatcher* peuvent traiter des arguments, définis par le biais de l'inspecteur de l'objet.