

22: Concevoir des équations

introduction

Ce tutoriel couvre la création d'équations: des énoncés mathématiques qui évaluent une expression en fonction de valeurs d'entrée. La clé des équations de Max est l'objet *expr*, qui fournit une multitude de fonctions mathématiques et un mécanisme «d'écriture personnelle» pour développer des énoncés complexes.

Jusqu'à présent, nous avons créé des processus mathématiques assez complexes en utilisant des objets mathématiques individuels (+, *, etc.). Cependant, dans de nombreux cas, nous pouvons regrouper un certain nombre d'objets en une seule équation mathématique, éliminant ainsi une grande partie de complexité inutile dans nos patches. L'outil utilisé pour cette simplification est l'objet *expr*, qui utilise une syntaxe spécifique pour créer une équation qui accepte plusieurs entrées et peut produire des résultats très complexes.

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

Aperçu du patch de dessin

Jetez un coup d'œil au patch du tutoriel pour voir le travail qui nous attend. Ce patch est divisé en deux parties encapsulées de la logique du patcheur et une routine de dessin pour un objet *lcd*, également encapsulée. La communication entre les parties est réalisée en utilisant des objets *value* et de paires d'objets *send* et *receive*. Pour voir le patch en action, activez le *metro* situé à gauche du patch à l'aide de la boîte *toggle* verte, puis appuyez sur la barre d'espace pour déclencher les événements situés en haut du patch. Le résultat variera de petits gribouillis aux dessins au traits et cercles abstraits de grand format. Chaque fois que vous appuyez sur la barre d'espace, l'écran *lcd* s'effacera et un nouveau dessin associé commencera à apparaître.

La boîte de *commentaire* en haut du patch montre trois lignes d'équations mathématiques:

$$x = \sin(A * y) - z * \cos(B * x);$$

$$y = z * \sin(C * x) - \cos(D * y);$$

$$z = E * \sin(x);$$

Ces expressions calculent trois valeurs (*x*, *y* et *z*) en appliquant des fonctions trigonométriques simples à leurs états *précédents*, ainsi que cinq valeurs fixes (*A*, *B*, *C*, *D* et *E*). Lorsqu'un tracé a lieu sur l'objet *lcd* (avec *z* visualisé sous la forme d'un cercle de taille variable), on peut voir qu'une grande variété de motifs peut être créée. Cette équation génère ce que l'on appelle un *attracteur étrange* - essentiellement une fonction itérée où les sorties précédentes de l'équation se combinent avec cinq variables (appelées coefficients) pour générer une variété de formes présentant un comportement chaotique.

- Double-cliquez sur l'objet *patcheur* **Do_the_math** pour l'ouvrir.

L'objet *send* nommé *draw* envoie des messages **bang** à la routine de dessin du subpatch **Draw_shape**. Cela déclenche les valeurs *x*, *y* et *z* nouvellement calculées, qui sont mises à l'échelle, regroupées dans des messages de commande, puis transmises par un objet *send* à l'objet *receive* attaché à l'écran *lcd*, qui exécute les commandes de dessin générées.

- Double-cliquez sur l'objet *patcheur* **Generate_coefficients** pour l'ouvrir.

Regardez en haut à gauche du patch principal. Lorsque l'objet *key* génère un **32** (ce qui signifie que la barre d'espace a été enfoncée), un *button* est utilisé pour générer un message **bang** envoyé au subpatch **Generate_coefficients**. Celui-ci effectue à son tour sept actions dans l'ordre, de droite à gauche: la génération de **5** valeurs *random* (*E* à *A*) qui vont de **-4,0** à **4,0**, la transmission d'un message **clear** à l'objet *lcd* et une réinitialisation de la valeur contenue dans les objets *value x, y* et *z*. En fait, la barre d'espace est la touche de **réinitialisation** globale de ce patch, en ce sens que pratiquement tous les segments du patch sont affectés lorsque cette touche est activée.

Utilisation d'*expr* pour la création d'équation

- Double-cliquez sur l'objet *patcheur* **Do_the_math** pour l'ouvrir.

Examinons maintenant les objets *expr* eux-mêmes. Bien qu'ils semblent compliqués, l'argument de chacun des objets *expr* est une équation relativement simple. La chose importante à réaliser est que les références **\$** se réfèrent simplement à un message entrant, avec son type de message et son nombre d'entrée. Ainsi, par exemple, l'utilisation d'une référence nommée **\$ f1** signifie que l'équation doit utiliser un message entrant (**\$**) qui est un nombre à virgule flottante (**f**) extrait de l'entrée **1** (**1**). Une fois que vous réalisez que la complexité apparente de l'équation est simplement sa référence aux messages entrants, le format de l'instruction *expr* est un peu plus compréhensible.

Les opérations utilisées dans nos instructions *expr* sont limitées aux fonctions *sin* (fonction sinus), *cos* (cosinus) et *** (multiplier); Cependant, il existe des dizaines d'opérations disponibles. Vous pouvez utiliser les opérateurs mathématiques courants (+, -, * et /), la logique (&&, ||), la comparaison par bit (&, |) et d'autres fonctions mathématiques du langage C telles que *abs*, *sin*, *cos* et *pow*. La documentation fournit une liste complète des fonctions disponibles.

Dans le cas de notre patch de tutoriel, les trois objets *expr* utilisent huit variables différentes (*x, y, z, A, B, C, D, E*) pour créer les trois valeurs cibles (*x, y, z*). Il y a ici une rétroaction intéressante: les variables cibles ne sont pas seulement utilisées pour la routine de dessin, mais aussi pour le prochain cycle de calcul. Les cinq variables *A* à *E* ne changent que lorsqu'on appuie sur la barre d'espace. Elles servent donc de point d'ancrage à la routine de dessin tandis que les variables *x, y* et *z* en constante évolution déterminent les dessins individuels de lignes et de cercles pour chaque cycle de calcul.

Ajouter votre propre équation

Pour rendre le dessin un peu plus intéressant, faisons un peu d'équation créative et utilisons-la pour générer une couleur de premier plan constamment changeante. Nous allons exploiter les objets *value x, y* et *z* qui sont mis à jour à chaque coup de *metro*.

Double-cliquez sur l'objet *patcheur* nommé **change_color** au bas du patch. Nous pouvons voir que la couleur de premier plan de l'écran *lcd* est modifiée par un trio d'objets *drunk* qui sont regroupés dans un message **frgb** envoyé à l'écran *lcd*. Ici, nous allons utiliser un objet *expr* pour calculer la fonction mathématique *exp* (*e* à la puissance de *x*). Si vous n'êtes pas familiarisé avec les propriétés de *e*, ce n'est pas grave. Il suffit de savoir qu'elle est de nature exponentielle et que sa valeur augmentera rapidement à mesure que la valeur d'entrée augmente.

Supprimez les objets *drunk* ou déconnectez-les de l'entrée et des objets *pack* et éloignez-les. Créez trois nouveaux objets *value* avec *x, y* et *z* comme arguments respectifs et connectez l'objet *entrée* à leurs entrées.

Maintenant, créez un nouvel objet et tapez ce qui suit:

```
expr int (exp ($ f1) * 5.)
```

En examinant l'argument / équation de l'intérieur vers l'extérieur, nous constatons que nous prenons une valeur à virgule flottante entrante (**\$ f1**), en utilisant la fonction **exp** pour créer une nouvelle valeur (**exp (\$ f1)**), en la multipliant par 5,0 (*** 5.**) Puis que nous la transformons en un nombre entier (**int**). Le résultat sera un nombre se situant approximativement dans la plage de valeurs de couleur (0-255), mais qui est majoritairement bas dans cette plage de nombres. Copiez cet objet *expr* deux fois, puis connectez la sortie des trois objets *value* (x, y et z) à chacun des objets *expr*. Connectez les sorties des objets *expr* dans le *pack* et voyez ce qui se passe.

Quand nous actionnons le *metro* et effaçons l'écran *lcd*, nous constatons que le dessin est coloré, mais qu'il tend vers les couleurs les plus sombres. Comme la coloration est influencée par les coordonnées *x*, *y* et *z* calculées, les couleurs auront tendance à être plus claires vers les bords extrêmes de la fenêtre de dessin. Il s'agit d'un exemple de calcul de commandes de dessin qui pourraient être difficiles avec la programmation traditionnelle basée sur les objets, mais qui sont très faciles à utiliser avec *expr*.

Résumé

L'utilisation de l'objet *expr* nous donne la possibilité de créer des équations mathématiques et logiques complexes sans avoir à recourir à de gros segments de programmation d'objets. Le grand nombre d'opérations pouvant être effectuées, combiné à la possibilité de définir et d'utiliser des entrées multiples, permet d'obtenir des résultats intéressants qui pourraient autrement être difficiles à créer.