

## 10: Dessin aléatoire

### introduction

Dans ce tutoriel, nous explorerons plus en détail le dessin à l'aide de l'objet *lcd*. Cependant, au lieu de dessiner explicitement des formes, nous allons rechercher des moyens de générer automatiquement des commandes de dessin utilisant les objets *random* et *drunk*.

De nombreuses formes d'art différentes utilisent des techniques aléatoires et pseudo-aléatoires pour générer un contenu intéressant ou inattendu. Nous verrons comment utiliser l'objet *random* pour produire une plage de nombres pour le matériau source, puisqu'il s'agit de la méthode utilisée dans Max pour créer un matériau aléatoire. L'objet *drunk* fournit un moyen d'effectuer une **marche aléatoire** à travers une plage de nombres, ce qui peut fournir l'imprévu de la génération de nombres aléatoires tout en évitant de grands sauts indésirables dans le nombre généré.

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

### Dessin avec random

Jetez un coup d'œil au patcheur du tutoriel. En haut à gauche du patcheur se trouve un patch qui utilise l'objet *random* (coloré en vert) pour générer un nombre aléatoire. Lorsque nous appuyons sur le *bouton* qui lui est relié, la sortie de l'objet est un nombre compris entre **0** et **999** – ce qui nous donne une plage de **1 000** nombres aléatoires potentiels. C'est le but de l'argument d'objet; il définit la plage de nombres que l'objet peut générer.

Chaque fois que vous cliquez sur le *bouton*, l'objet *random* génère un autre nombre. Si vous souhaitez modifier la plage de la sortie, vous pouvez modifier l'argument - ou vous pouvez envoyer une nouvelle valeur dans l'entrée droite de l'objet *random*. Si nous entrons **20** dans la boîte de *nombre* connectée en haut, nous pouvons voir que le fait d'appuyer sur le *bouton* va maintenant générer des nombres compris entre **0** et **19**. A présent, utilisons ceci pour générer des graphiques intéressants dans l'objet *lcd* en bas à gauche de la fenêtre.

L'objet *patcheur* étiqueté **exemple 1** est un générateur graphique automatisé pour l'écran *lcd* ci-dessous. Un objet *metro* fournit un flux continu de messages bang dans cinq objets aléatoires, chacun avec une plage appropriée à la taille de l'objet *lcd* (**320** pixels de large et **240** pixels de haut). La sortie des deux premiers objets *random* est manipulée avec des objets mathématiques pour créer les nombres dont nous avons besoin. Tout est empaqueté pour créer une liste de sept nombres (via l'objet *pack*); le message **paintoval** est placé avant la liste (via l'objet *prepend*), puis envoyé à l'écran *lcd*.

Afin de voir ce que les sept chiffres de la liste représentent, nous devons vérifier le texte de référence du *lcd*. Nous pouvons le faire en déverrouillant le patcheur, en sélectionnant l'objet *lcd*, puis en choisissant **Référence** dans le menu **Aide**. Ceci affiche la page du manuel *lcd*, où nous pouvons examiner le message **paintoval**.

Selon le manuel de référence, les arguments sont les suivants : gauche, haut, droite, bas et couleur (envoyés comme une liste de trois valeurs représentant les quantités de rouge, vert et bleu de la couleur à utiliser). En regardant le code de dessin automatisé, nous constatons que le premier objet *random* fournit l'emplacement gauche et droit (avec 5 pixels ajoutés et soustraits pour obtenir une forme de 10 pixels de large); le deuxième objet *random* est similaire, mais il fournit les emplacements haut et bas de la forme. Enfin, les trois derniers objets *random* créent trois nombres

aléatoires (compris entre **0** et **255**) qui créeront une couleur RVB aléatoire pour le dessin.

Lorsque nous activons le *metro* avec le *toggle*, nous voyons que *lcd* se remplit rapidement de petits cercles (10 pixels) de couleur aléatoire. C'est un excellent exemple d'emplacement et de sélection de couleur véritablement aléatoires. Mais que faire si nous voulons exercer un peu plus de contrôle sur l'emplacement du dessin ?

### *Dessin avec drunk*

Il existe de nombreux types de comportements aléatoires qui peuvent être modélisés par l'ordinateur. L'objet *random* fournit l'approximation la plus proche de l'aléatoire "pur" que nous puissions obtenir; chaque nombre dans son plage a une probabilité égale d'être sélectionné. Une alternative à l'objet *random* est l'objet *drunk*, ainsi appelé parce qu'il vous permet d'effectuer la «marche ivre» à travers une plage de nombres; cette randomisation (techniquement une variante de ce qu'on appelle le **mouvement brownien**) caractérise de nombreux processus "aléatoires" dans la nature.

Juste en dessous de notre exemple *random* (en haut à gauche du patch) se trouve un deuxième patch de test pour l'objet *drunk*. Si vous cliquez sur le *bouton* connecté, vous verrez qu'il nous donne un nombre aléatoire basé sur la plage fournie par son premier argument (un argument de **1000** nous donne des nombres compris entre **0** et **999**). Le fait d'appuyer une seconde fois sur le *bouton* donne cependant un résultat quelque peu différent: nous obtenons toujours un nombre aléatoire, mais il reste dans la plage de +/- **20** de la valeur précédente. Continuez à appuyer sur le *bouton*, et vous verrez que la valeur restera toujours dans un intervalle de **20** du nombre choisi précédemment. Le deuxième argument de *drunk* nous donne la taille maximale du pas pour chaque réception d'un **bang**.

Comme pour l'objet *random*, les deux arguments de *drunk* peuvent être modifiés en connectant des boîtes de *nombres* aux entrées prévues à cet effet. Dans le cas d'un objet *drunk*, la deuxième entrée modifie la valeur de la plage, tandis que l'entrée droite (la troisième) modifie la valeur de pas. Si nous modifions la plage et les tailles de pas par de plus petits nombres (**10** et **2** respectivement), nous pouvons voir comment la sortie est réduite à une plus petite plage lorsque nous cliquons à nouveau sur le *bouton*.

Nous pouvons effacer le canevas de l'écran *lcd* en cliquant sur la boîte de message intitulée Effacer juste au-dessus de l'écran *lcd* dans le patch du didacticiel. L'objet patcheur intitulé **Exemple 2** montre comment utiliser l'objet *drunk* pour effectuer une autre tâche de dessin automatisée. En apparence, cela ressemble beaucoup à la routine de dessin que nous avons utilisée pour démontrer l'objet *random*, simplement avec un objet substitué à un autre. Après avoir effacé l'objet *lcd*, lancez le *metro* en cliquant sur le *bouton* situé au-dessus. Des messages **paintrect** sont envoyés à l'objet *lcd* avec une liste de valeurs entières générées; les rectangles résultants sont dispersés sur l'écran *lcd*, mais ne font jamais de grands bonds, car ils sont contraints par la petite taille de pas (20 x 15 pixels) des objets *drunk*. Vous remarquerez également que la couleur change très lentement, car les arguments de couleur créés sont également contraints par la taille de pas (**5**) des objets *drunk*.

### *Plus de dessin random*

Le troisième ensemble d'objets de dessin automatisé se trouve dans l'objet *patcher* nommé **Exemple 3**. Cette section du patch semble un peu plus compliquée et utilise un nouveau message de dessin pour l'objet *lcd*: **framepoly**. Si nous ouvrons la page de référence *lcd*, nous voyons que ce message attend une série de paires x / y pour créer un cadre de polygone, où chaque paire x / y représente un coin ou un sommet. Notre routine automatisée utilise huit objets *random* pour créer quatre paires de

coordonnées. Celles-ci sont combinées dans une liste (avec *pack*), précédées du message **framepoly** (avec *prepend*), puis envoyées à l'écran *lcd*. Plutôt que d'intégrer le message **framepoly** aux informations sur les couleurs, nous calculons séparément un ensemble aléatoire de trois valeurs, les regroupons en une liste et créons un message **frgb** (pour **RVB au premier plan**), qui est également envoyé au *lcd*.

Cette routine adopte une approche différente de la sélection aléatoire de points. Plutôt que de sélectionner n'importe quel point de la surface de l'écran *lcd*, les nombres aléatoires génèrent une valeur comprise entre 0 et 9, puis *multiplient* le résultat pour compléter la sélection des coordonnées. Dans le cas de coordonnées horizontales (x), la valeur est multipliée par **32**; pour les coordonnées verticales (y), la valeur est multipliée par **24**. Ainsi, la surface de l'écran *lcd* est divisée en une grille de 10 x 10 et toutes les coordonnées sont "verrouillées" dans cette grille.

Lorsque nous activons le *metro* pour cette section du patch (via l'objet *toggle* situé au-dessus de celui-ci), nous constatons que le cadre du polygone est dessiné dans la couleur générée (aléatoire), mais nous voyons aussi rapidement le résultat de la limitation de la sélection de nombres aléatoires. : le graphique prend une qualité semblable à celle du tissu, particulièrement visible sur les bords. En modifiant légèrement la façon dont nous utilisons l'objet *random*, nous avons exercé un certain contrôle sur le résultat généré.

Un résultat légèrement différent peut être observé si nous remplaçons les objets *random* utilisés pour la création de coordonnées par des objets *drunk*. Remplacez tous les **10** objets aléatoires par **10 2** objets *drunk*. Maintenant, lorsque vous exécutez le patch, la création du tissu semble encore plus contrôlée. Vous pouvez affiner la fonction de dessin en remplaçant les générateurs de couleur, en modifiant tous les objets *random 255* en quelque chose comme *drunk 255 15*. La couleur changera alors de manière fluide, ce qui vous donnera une perspective différente sur le dessin généré.

## Résumé

Nous en avons appris un peu plus sur le dessin dans l'objet *lcd*, en utilisant les messages **paintoval** (pour créer de petits cercles), **paintrect** (pour créer de petits carrés) et **framepoly** (pour créer des lignes jointes). Plus intéressant encore, nous avons également appris à utiliser Max pour effectuer le dessin, en utilisant les objets *random* et *drunk* pour générer des coordonnées et des couleurs sur la surface de l'écran *lcd*. Nous avons également constaté, à l'aide de plages et de multiplications, que nous pouvons manipuler la génération de nombres aléatoires pour répondre aux besoins spécifiques d'une application. Il ne faut pas s'étonner que les objets *random* et *drunk* soient au cœur de nombreuses applications de musique et de graphisme génératives.