17: Structures de données et probabilités

introduction

Dans ce tutoriel, nous étudions l'utilisation de l'objet *itable* pour le stockage, la transformation et le suivi des histogrammes. L'objet *itable* peut être dessiné ou défini à l'aide de diverses formules. Nous verrons comment utiliser l'objet *uzi* pour générer rapidement de nombreux messages **bang** ou des flux de nombres ascendants, tous deux pouvant être utilisés pour créer rapidement de nombreuses données générées par des algorithmes.

Une structure de données bidimensionnelle (comme *itable*) nous donne la possibilité de créer des transformations de données faciles à configurer. Ces ***fonctions de transfert*** vous permettent de créer des courbes irrégulières, des tableaux de probabilités qui pondèrent la sortie en fonction de l'entrée et des modèles aléatoires faciles à visualiser. De plus, l'introduction de l'objet *uzi* pour la génération rapide de messages procéduraux nous aide à trouver un moyen de créer les fonctions nécessaires pour charger ces objets *itable* de manière intéressante.

L'objet *itable* utilisé dans ce tutoriel est pratiquement équivalent à l'objet *table*, qui possède sa propre fenêtre et peut être saisi dans une boîte d'objet standard. L'*itable* est utilisée ici simplement parce qu'il apparaît intégré visuellement dans un patcheur plutôt que dans une fenêtre d'édition séparée.

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

Structures de données de base avec *itable*

Dans le patcheur de ce didacticiel, vous verrez quelques patchs d'affichage / dessin (étiquetés avec des nombres) et plusieurs petits calculs (étiquetés avec des lettres). Nous allons commencer à travailler avec la zone de patch intitulée **1**. Ce patch comporte un objet *itable* ; un objet qui fournit une interface visuelle à une structure de données bidimensionnelle appelée *table*. La structure *table* / *itable* nous donne une combinaison X / Y et une interface simple pour ces données : envoyez un nombre représentant un index sur l'axe des X et vous recevrez la valeur correspondante de l'axe des Y.

Lorsque vous ouvrez le patch pour la première fois, l'*itable* contient une distribution aléatoire de valeurs. Si nous déplaçons le curseur en haut du patch, il parcourt les valeurs de l'*itable* et émet la valeur Y de sa sortie gauche pour la position X représentée par le curseur. Nous avons connecté cela à une boîte de *nombres* et un *curseur*, afin que nous puissions voir les résultats de la fonction de transfert décrite par l'*itable*. A côté du *curseur* d'entrée se trouvent une paire de boîtes de *messages* avec des listes de deux nombres entiers; celles-ci définiront l'emplacement du premier nombre entier (X) à la valeur du deuxième nombre entier (Y). Ainsi, avec la première boîte de *message*, la valeur à l'emplacement **50** sera fixée à **80**. Si nous cliquons sur cette boîte de *message*, puis déplaçons le *curseur* (ou la boîte de *nombres*) sur **50**, nous pouvons voir que la sortie de la variable *itable* est de **80**. De même, après avoir cliqué sur la deuxième boîte de *message* (**100 20**), nous pouvons voir qu'une valeur d'entrée de **100** génère une valeur de **20**.

Le message ***clear*** "efface" la vue de l'*itable*, mais les valeurs sont toujours intrinsèques à l'*itable* ; lorsqu'il est effacé, l'*itable* attribue simplement une valeur **0** à chaque point de données possible. Une autre façon de modifier les valeurs de *l'itable* est de dessiner à l'intérieur de *l'itable* lui-même. Si vous placez la souris à l'intérieur de *l'itable* et cliquez-glissez la souris, vous pouvez dessiner le contenu de la fonction que vous désirez. Si vous devez tracer des lignes droites, vous pouvez

maintenir la touche Maj enfoncée pendant que vous déplacez la souris, puis cliquez sur la souris pour valider la nouvelle ligne. L'utilisation de ces commandes de dessin vous offre un moyen facile de créer manuellement des fonctions de transfert intéressantes. (Notez cependant qu'il n'existe qu'une seule valeur Y pour tout X. Vous ne pouvez pas dessiner une ligne qui se chevauche verticalement.)

Sous le patch **1**, vous trouverez trois petits patchs qui utilisent des objets *uzi* pour générer tous les points de la variable *itable*. Si vous cliquez sur le bouton **A**, une fonction est créée où la valeur d'entrée est identique à la valeur de sortie. En cliquant sur le bouton **B**, on obtient l'inverse: les valeurs d'entrée comprises entre **0** et **127** sont converties en une plage de **127** à **0**. Le bouton **C** produit une distribution aléatoire similaire au contenu de départ de l'objet *itable*.

Créer des formules avec *uzi*

Examinons de plus près ces routines générant des fonctions, en particulier l'objet *uzi*. Comme son nom l'indique, l'objet *uzi* est conçu pour déclencher rapidement des messages **bang**. Un seul **bang** à l'entrée d'un objet *uzi* générera autant de messages **bang** que spécifiés dans l'argument de l'objet ; par conséquent, *uzi* **128** enverra **128** messages **bang** à partir de la sortie de gauche. Afin de garder une trace du nombre de **bang** actuel, la sortie ***la plus à droite*** génère un décompte (en commençant par **1**) - cette sortie est ce que nous utilisons pour l'index de l'objet *itable*. Dans le segment de patch **A**, ce nombre est réduit de **1** (il commence donc par **0**, le début de la plage *itable*) et est utilisé pour les valeurs X et Y, ce qui nous donne la sortie en ligne droite pour une fonction.

Le segment **B** est similaire, sauf que la valeur Y utilise l'objet *!* - pour soustraire la valeur entrante à **127**. Il en résulte une rampe similaire, mais dans la direction opposée - les valeurs X les plus faibles ont les valeurs Y les plus élevées. Le segment **C** utilise un objet *random* pour créer des valeurs aléatoires dans la liste de sortie. Il doit utiliser l'objet *swap* (qui "échange simplement" les valeurs gauche et droite qu'il reçoit) pour inverser l'ordre du contenu de la liste afin que la sortie de l'objet *random* soit utilisée pour la valeur Y au lieu de la valeur X.

Modifions un peu le segment **B** pour voir comment différentes fonctions pourraient être créées. Si nous modifions l'objet *!* - en un objet division *(/)* avec un argument de **4**, nous réduirons la plage des valeurs Y à seulement 1/4 de la plage verticale. De même, si nous multiplions *(*)* la valeur par **2**, nous atteindrons le maximum de la plage des Y à mi-chemin des valeurs X. Évidemment, avec un peu de manipulation, vous pouvez utiliser *uzi* et un peu de maths pour créer de nombreuses fonctions complexes.

Histogrammes avec *table*

La zone de patchs de droite (appelée **2**) constitue une utilisation intéressante du système *itable*, ainsi que plusieurs patchs de formule et un nouvel objet appelé *histo*. L'objet *histo* remplit une fonction d'histogramme : il enregistre le nombre de fois où un nombre est entré dans son entrée. Chaque fois qu'une valeur entre dans l'entrée, l'objet produit le nombre de fois où elle a été reçue par sa sortie droite, suivie immédiatement par la valeur elle-même par sa sortie gauche. Elle peut ensuite être directement introduite dans l'objet *itable*, où l'entrée gauche devient le X et l'entrée droite devient le Y. Si vous utilisez le *curseur* situé en haut du patch pour "scratcher" dans de nombreux nombres, vous verrez l'objet *itable* afficher la distribution relative des valeurs que vous avez envoyées. Si vous souhaitez effacer les objets *itable* et *histo*, vous pouvez appuyer sur la touche de retour de votre clavier (ASCII 13).

Créer des distributions de probabilité

Comme avec notre premier patch de dessin, nous disposons d'un certain nombre de zones de patch plus petites qui peuvent être utilisées pour générer des fonctions dans l'objet *itable*. Cependant, au lieu de dessiner des fonctions discrètes, celles-ci génèrent des distributions floues en envoyant des valeurs aléatoires et semi-aléatoires à l'objet *histo* ; cet histogramme est ensuite alimenté en *itable* pour générer une carte de probabilité. Les segments **D** et **E** utilisent respectivement *random* et *drunk* pour générer les valeurs. En cliquant à plusieurs reprises sur les objets *button* attachés, vous verrez comment chacun de ces effets affecte la courbe de distribution : *random* touche toutes les valeurs de manière assez uniforme, mais *drunk* est beaucoup plus irrégulier dans sa distribution. De nouveau, nous pouvons effacer l'histogramme à tout moment avec la touche Retour.

- Ouvrir le subpatch **More_ops**

Les segments de patch **F** et **G** adoptent une approche quelque peu différente. Le segment **F** génère deux nombres aléatoires, puis utilise l'objet *minimum* pour choisir le plus petit des deux nombres. Cela aura tendance à mettre en valeur les nombres les plus bas de la plage; en cliquant de façon répétée sur le *button* connecté, les nombres les plus bas seront utilisés plus souvent que les valeurs les plus élevées. Le segment **G** fait l'inverse: il utilise le plus grand des deux nombres aléatoires en les plaçant dans l'objet *maximum*. Ainsi, un clic répété sur le *button* connecté aura tendance à générer plus de nombres élevés.

Enfin, le segment **H** crée une moyenne de 3 nombres aléatoires en générant les nombres, en les additionnant, puis en divisant la somme par **3**. Si vous effacez l'*itable* et vous cliquez de manière répétée sur le *button* **H**, vous verrez apparaître une courbe en forme de cloche se générer, puisque la moyenne des trois nombres aléatoires aura tendance à se situer au milieu de la plage. Il s'agit d'un type simple de distribution gaussienne (ou normale), que l'on retrouve couramment dans les recherches statistiques lorsque la plupart des sujets se situent au milieu d'une plage de valeurs possibles.

Utilisation de la distribution de probabilité

Activez l'objet *metro* à l'aide du *toggle* situé en haut du patch **2** ; cela entraînera l'envoi de messages **bang** à l'objet *itable*. Cela produit une sortie «quantile» aléatoire à générer - une valeur de sortie aléatoire mais pondérée de l'objet *itable*. Le résultat est que la zone *lcd* sous *l'itable* commencera à dessiner de petits cercles où la densité est à peu près équivalente à la courbe de distribution montrée dans la routine *itable*. Si vous souhaitez effacer le *lcd*, vous pouvez appuyer sur la barre d'espace (ASCII **32**) pour relancer le dessin. (L'emplacement en **Y** de chaque « dort » étant aléatoire, la courbe de probabilité apparaît dans la densité de gauche à droite.)

Si vous laissez le *metro* en marche tout en utilisant les générateurs de fonctions *uzi* de droite, le dessin devrait simuler les courbes de distribution de l'objet *itable*. Pour tester des courbes plus radicales, vous pouvez tracer des lignes nettes au sein de l'objet *itable* pour forcer certaines courbes de sortie; vous pouvez également utiliser le *curseur* pour «ombrer» certaines parties de l'équation, ce qui oblige la sortie du dessin s'ombrage par segments en réponse à la nouvelle courbe de distribution.

Résumé

De nombreux processus différents dépendent de la possibilité de créer des fonctions qui transforment une plage d'entrée en une plage de sortie. Dans les tutoriels précédents, nous avons vu comment l'objet *scale* pouvait le faire pour nous. Cependant, si nous voulons modifier

manuellement la distribution des valeurs ou si nous voulons utiliser une formule pour la transformation de données, nous devons utiliser un système plus complexe. Comme nous l'avons vu dans ce tutoriel, les objets *table* / *itable* répondent parfaitement à cet objectif, nous permettant de fabriquer ou de dessiner manuellement une fonction de transfert.

En utilisant l'objet *histo*, nous avons également vu comment nous pouvons générer le type de courbes de distribution qui servent de techniques aléatoires courantes dans l'art génératif et la musique stochastique. Qu'il s'agisse du flou de la véritable génération aléatoire ou des tendances que l'on trouve avec des calculs plus discrets, ces courbes peuvent être utilisées pour générer et guider les résultats vers un résultat souhaité.